


# Nonlinear Optimization of Light Field Point Cloud

Yuriy Anisimov<sup>1,2,\*</sup> , Jason Raphael Rambach<sup>1</sup>  and Didier Stricker<sup>1,2</sup>

<sup>1</sup> Department of Augmented Vision, German Research Center for Artificial Intelligence, Trippstadter Str. 122, 67663 Kaiserslautern, Germany; Jason.Rambach@dfki.de (J.R.R.); Didier.Stricker@dfki.de (D.S.)

<sup>2</sup> Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany

\* Correspondence: Yuriy.Anisimov@dfki.de

**Abstract:** The problem of accurate three-dimensional reconstruction is important for many research and industrial applications. Light field depth estimation utilizes many observations of the scene and hence can provide accurate reconstruction. We present a method, which enhances existing reconstruction algorithm with per-layer disparity filtering and consistency-based holes filling. Together with that we reformulate the reconstruction result to a form of point cloud from different light field viewpoints and propose a non-linear optimization of it. The capability of our method to reconstruct scenes with acceptable quality was verified by evaluation on a publicly available dataset.

**Keywords:** light field; depth estimation; point cloud

## 1. Introduction

A definition of a light field was first given by Gershun in [1]. The light field is formed from all light rays, which are passing through all points in space in all directions. It was generalized in the work of Adelson and Bergen [2], where the sufficiency of finite light rays sampling was stated. For computer vision tasks, based on the two-plane parameterization from [3], light field can be considered as a one- or two-dimensional set of two-dimensional images, called light field views, and captured with the preservation of fixed physical distance between them.

Various devices can be used for light field acquisition. For static scenes, an ordinary perspective camera can be moved on certain distances for capturing a scene from multiple viewpoints. In cases when ensuring of equal camera movements is not possible, a control pattern can be used with two oppositely oriented cameras [4]. For dynamic scenes and varying resolutions, two different configurations can be used.

A light field camera can consist of multiple isolated camera sensors and lenses [5] or of one camera sensor and multi-lens arrays in front of it [6]. Such an approach evolves to the micro-lens array, where the light field images are captured as the small views from many viewpoints [7]. Modern configurations can be downscaled to the form-factor of mobile cameras [8].

Various features of light field images attract the attention for different research and industrial applications. For instance, digital refocusing allows changing the focus of the captured image [7]. In addition, light fields can be used as a source for the accurate novel view synthesis [9].

Additionally, the presence of multiple views in the light field can be used for the scene three-dimensional reconstruction. One important feature of the light field cameras for that purpose is related to the view alignment. Capturing units of light field camera are oriented in the same direction and the distance between them is known and fixed, which allows simplifying the search of matching correspondence among light field views.

In this work, we present an extension of our depth estimation algorithm from [10]. An example of the result of our algorithm is demonstrated in Figure 1. The contributions compared to this algorithm are:



**Citation:** Anisimov, Y.; Rambach, J.; Stricker, D. Nonlinear Optimization of Light Field Point Cloud. *Sensors* **2022**, *22*, 814. <https://doi.org/10.3390/s22030814>

Academic Editor: Denis Laurendeau

Received: 20 December 2021

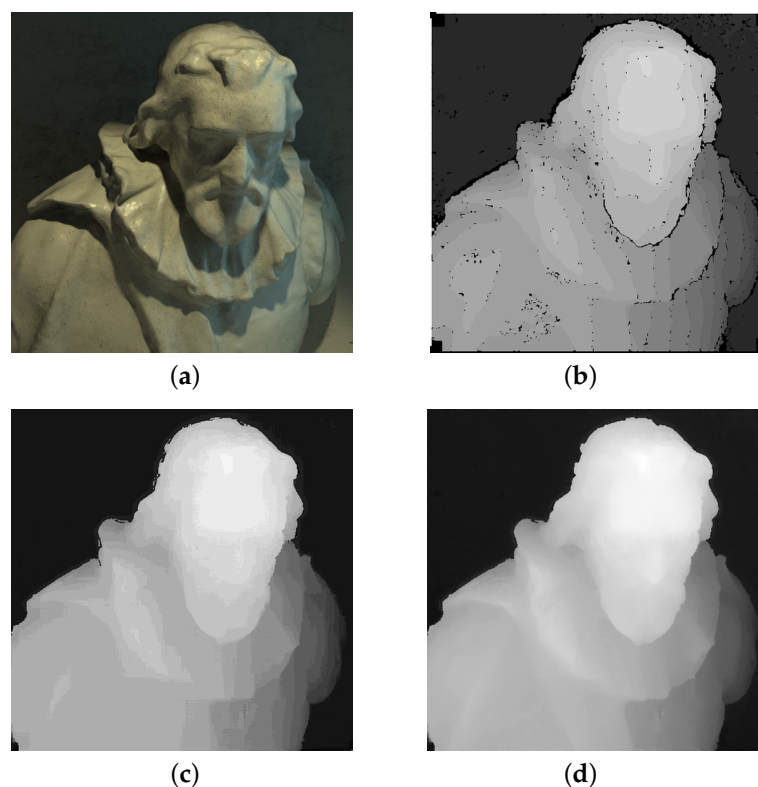
Accepted: 18 January 2022

Published: 21 January 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).



**Figure 1.** Results of our algorithm for a synthetic scene "cotton" from 4D Light Field Benchmark [11]: (a) center image from a light field, (b) initial disparity map, (c) final disparity map, (d) refined disparity map.

- per-layer disparity filtering and color-consistency based holes filling, which noticeably improves the accuracy of initial disparity map,
- different representation of the reconstruction result in the form of point cloud, with the additional step of its nonlinear refinement over light field views.

## 2. Related Work

Many methods exist for solving the light field depth estimation task. Wanner and Goldluecke in [12] propose the labeling of epipolar-plane images (EPIs) [13] with variational methods, introducing the structure tensor for its analysis. Neri et al. in [14] provides a combination of multi-resolution multi-view stereo and variational methods for solving depth estimation tasks. A method which takes pixel occlusions into consideration was proposed by Wang et al. in [15].

Johannsen et al. in [16] construct the light field dictionaries with specific disparity values based on EPIs. Strecke et al. [17] estimate depth and normals using partial focal stacks with their joint optimization. Preceding methods [10,18] used a construction of matching cost volume in the bordered space with further cost refinement.

In the method of Shin et al. [19], a multi-stream architecture of EPI analysis for depth estimation is presented. Huang et al. [20] proposed a model for disparity estimation based on multi-scale cost aggregation with additional edge guidance.

Few studies have been published on the point cloud utilization for light fields particularly. In their analysis, Perra et al. [21] show the object extraction and point cloud estimation from depth maps together with a comparison of point clouds, retrieved from the two popular plenoptic cameras. In the work of Ferreira et al. [22], the RANSAC method is applied to SIFT-based features from plenoptic images for estimating the virtual point cloud. This point cloud is back-projected to the micro-lenses space and further optimized using least squares.

An approach of Farhood et al. [23] shows how the depth map, obtained by the light field camera, can be improved for getting the high-quality point cloud. The depth map is modified by histogram manipulations, aimed for better separation of depth layers, and then further enhanced by adding information about the objects' edges. This provides better separation of different objects in a point cloud. In the method of Yucer et al. [24], the point cloud is reconstructed from patch-based local light field gradient information.

Light field point cloud estimation for the case of unfocused light field can be considered as the extension of stereo to multi-view with strict baseline constraints. A classical work in this direction was published by Liu et al. [25]. The estimation process contains the detection of initial point clouds by stereo matching, their merging with downsampling and further mesh generation. The construction of experimental camera, used in this publication, can be considered as a combination of sparse focused light field cameras due to capturing devices placement.

In the last few years, many approaches have been utilizing deep learning methods for point cloud reconstruction. A noticeable publication in this direction was published by Chen et al. [26] and presents a two-stage method of multi-view point cloud estimation. First, the coarse depth map is estimated by using MVSNNet [27]. It is used as a component for loss estimation and as an initial estimation for the point cloud. This point cloud is augmented based on image feature pyramid, extracted from the provided views; and iteratively refined based on the PointFlow network, proposed in that paper.

### 3. Initial Disparity Map Generation

This section describes the steps for getting an initial disparity map, which is used afterwards as an initialization step for the further nonlinear optimization. For that, we follow and improve the disparity estimation algorithm, described in [10].

#### 3.1. Light Field Parameterization

A classical way of representing the light rays was defined by Adelson and Bergen in [2]. They were parameterizing rays by a plenoptic function, consisting of three dimensions for the ray position and two dimensions for its orientation. However, this description might not be very convenient for the utilization in computer vision algorithms due to its complexity.

One of the common descriptions of light field exists in a form of two-plane parameterization, proposed by Levoy and Hanrahan in [3]. Following this definition, every ray of a light field is described by the intersection over a plane of spatial coordinates  $(u, v)$  and an angular coordinates plane  $(s, t)$ , as demonstrated in Figure 2. We denote the light field as  $L$ , with a specific ray projection as  $L(u, v, s, t)$ .

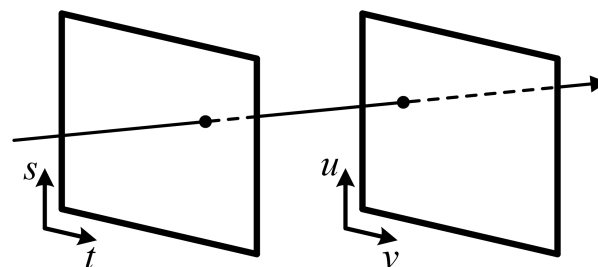


Figure 2. Two-plane light field parameterization from [3].

#### 3.2. Matching Between Light Field Views

In the context of computer vision, the light field is considered as a set of two-dimensional images assembled into a two-dimensional array. Individual rays of the light field are projected onto these images in the form of pixels. To recover the depth information associated with each light field ray, we use pixel similarity measurement techniques. It is convenient to use a concept of "disparity", which is inversely proportional to the depth, as disparity is

expressed as a distance in pixel units and can be calculated explicitly from the light field views. Based on the two-plane parameterization, the matching position of a certain pixel  $(\hat{u}, \hat{v})$  from a given reference light field view  $(\hat{s}, \hat{t})$  with the disparity  $d$  can be found in arbitrary light field view  $(s, t)$  as [28]:

$$p(u, v, s, t, d) = L(\hat{u} + (\hat{s} - s)d, \hat{v} + (\hat{t} - t)d, s, t). \quad (1)$$

### 3.3. Reference and Anchor Views

Our disparity estimation approach starts with the computing of coarse disparity maps for "reference" view from the "anchor" views of the light field. For reference, we select a view, which lies in the middle of both light field angular axes. As anchors, we define the views at the borders of the light field, which are lying on the cross with the reference view in its center. Figure 3 illustrates how the reference and anchor views are placed in the light field image. These views are important for the disparity estimation as they cover all visible scene points, projected to the light field image.

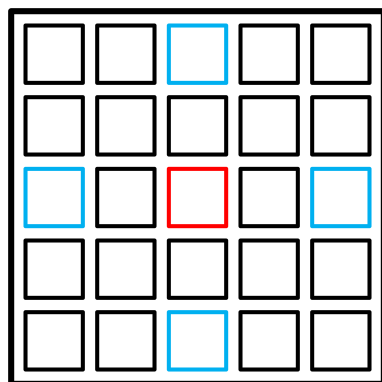


Figure 3. Reference (red) and anchor (blue) views of the light field.

To generate the set of coarse disparity maps, these views are utilized pairwise. Based on the reference light field view with coordinates  $(\hat{s}, \hat{t})$ , we define a set of four cross-lying views in the light field image  $V = \{(\hat{s}, t_{min}), (\hat{s}, t_{max}), (s_{min}, \hat{t}), (s_{max}, \hat{t})\}$ , where  $s_{min}, t_{min}, s_{max}, t_{max}$  corresponds to minimal and maximal possible indexes on vertical ( $s$ ) and horizontal ( $t$ ) angular dimensions of the light field image.

### 3.4. Matching Cost Generation

A matching cost can be described as a three-dimensional structure, where every element represents the comparison of every pixel in reference view with the corresponding pixel in another view based on the disparity hypothesis, lying in the range of  $T = [d_{min}, d_{max}]$ , where  $d_{min}$  and  $d_{max}$  are the minimum and maximum disparity hypotheses.

Different methods exist to measure pixel similarity. Roughly, these methods can be divided into pixel-wise and window-wise. Commonly used pixel-wise functions are Manhattan and Euclidean distances [29]. Widely used window-wise measures are the sum of absolute differences, the sum of squared differences and normalized cross-correlation [30]. Window-wise measures can provide more accurate results in contrast to pixel-wise methods, but the computation time increases since, for each pixel in an image, more pixels around are involved, which can limit usage on window-based approaches in rapid estimation algorithms.

A Census transformation was formulated in [31] as intensities-dependent non-parametric transform. The base Census estimation works as follows. For a pixel  $p$  with pixel coordinates  $(u, v)$  in an image  $I$ , its intensity value is compared with other pixels around the reference, coordinates of which are defined in a set  $M$ :

$$I_C(u, v) = \bigotimes_{[i,j] \in M} \xi(I(u, v), I(u + i, v + j)), \quad (2)$$

where  $\otimes$  stands for bit-wise concatenation, and pixel relations are defined as:

$$\zeta(v_1, v_2) = \begin{cases} 0, & v_1 \leq v_2 \\ 1, & v_1 > v_2 \end{cases} \quad (3)$$

Such estimations can be done in a dense way, taking all pixels within the window into consideration, or in a sparse way by defining the coordinates of specific pixels to be involved in the Census image. Originally, this function is considered to be applied to the single-channel image; in our work, it is extended to be performed on RGB images, treating every channel separately.

To compare values of Census pixels in different views and to generate the matching cost, the Hamming distance is used. For two images in Census-transformed light field  $L_c$  with coordinates  $(\hat{s}, \hat{t})$  and  $(s, t)$ :

$$C_c(u, v, d) = HD(p_c(u, v, s, t, d), L_c(\hat{u}, \hat{v}, \hat{s}, \hat{t})), \quad (4)$$

where  $HD()$  is the Hamming distance function: for two vectors,  $x_i$  and  $x_j$  ( $|x_i| = |x_j| = n$ , here and further  $|\dots|$  denotes cardinality), it can be determined as a sum of elements with different values:

$$HD(x_i, x_j) = \sum_{k=1}^n x_{ik} \oplus x_{jk}, \quad (5)$$

where  $\oplus$  stands for exclusive disjunction.

Figure 4 demonstrates the principles of Census transformation and Hamming distance estimation. For every view in  $V$ , cost is generated by matching between this view and the opposite view on the same axis. In general, matching costs collected from two images might suffer from a big amount of ambiguities, which will negatively affect the estimation of disparity map by introducing noise to the image. The generated matching cost needs additional optimization to make it usable for accurate disparity estimation with a low noise level.

### 3.5. Semi-Global Matching

To solve this issue, we use a widely known Semi-Global Matching (SGM) method, proposed in [32]. This method can be considered as the optimal one between local-only matching cost collection and the global cost optimization, which can provide the most accurate results, but with significant computational load.

For each pixel  $p = (u, v)$  and  $d \in T$ , after traversing in direction  $r$ , formulated as a two-dimensional vector  $r = \{\Delta u, \Delta v\}$ , aggregated cost  $L_r$  is

$$\begin{aligned} L_r(p, d) = & C(p, d) + \\ & \min(L_r(p - r, d), \\ & L_r(p - r, d - 1) + P1, \\ & L_r(p - r, d + 1) + P1, \\ & \min_t L_r(p - r, t) + P2), \end{aligned} \quad (6)$$

where  $P1$  and  $P2$  are penalty parameters for neighborhood disparities,  $P2 \geq P1$ . Costs are then summarized among all directions:

$$C_S(p, d) = \sum_r L_r(p, d). \quad (7)$$

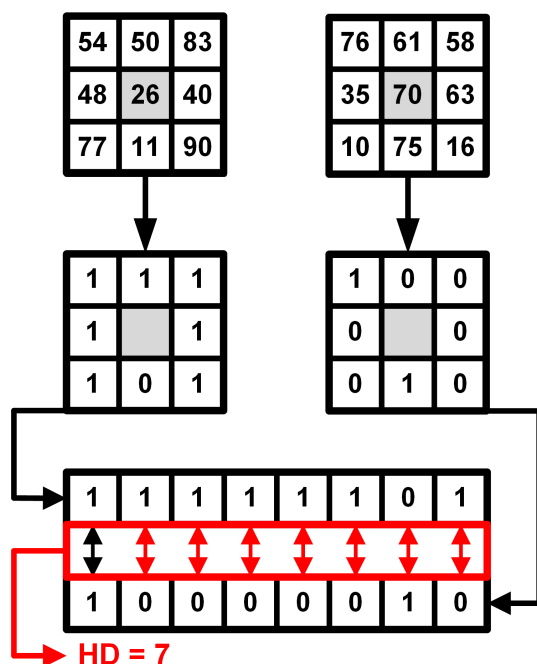


Figure 4. Visualization of image Census transformation and Hamming distance estimation.

### 3.6. Disparity Map Forming

From the matching cost, the disparity value for every pixel  $p$  can be estimated using the winner-takes-all (WTA) method as:

$$D_{V_i}(p) = \arg \min_d C_s(p, d). \tag{8}$$

Despite the matching cost being subject to SGM, some noise pixels can still be present. For that, a median filter applied, where for every disparity map pixel, the median value of its neighborhood is associated.

### 3.7. Consistency Checks and Merging

After performing the previously described steps, we obtain a set of four disparity maps  $D_V$ . Each of these maps is used for the consistency check. First, disparity maps are projected on a plane of reference view  $(\hat{s}, \hat{t})$ . To do so, the order of views in Equation (1) is modified from  $(\hat{s} - s)$  and  $(\hat{t} - t)$  to  $(s - \hat{s})$  and  $(t - \hat{t})$ , where  $s$  and  $t$  are related to the actual view position in the light field, and for every pixel  $d = D_{V_i}(u, v), i = 1 \dots |V|$ .

Next, for each reprojected  $D_{V_i}$ , we check the matching pixel in its opposite view  $D_{V_o}$  (based on the original views placement) for their inequality:

$$|D_{V_i}(u, v) - D_{V_o}(u, v)| < \varphi, \tag{9}$$

where  $\varphi$  stands for confidence threshold. Every pixel of the fused initial disparity map  $D_C$  is computed as the average of the corresponding pixel values from  $D_V$ , for which the condition presented in Equation (9) is met. The pixel is discarded as uncertain if this condition is not true.

### 3.8. Per-Layer Disparity Filtering

Quality of the disparity image obtained with Hamming distance matching may contain noise elements for individual pixels, which would be difficult to remove with a median filter. For that, we propose a filter, based on per-layer decomposition of the initial disparity

map. For every available disparity hypothesis  $d \in T$ , the new image, containing only pixels with disparity value  $d$ , is created:

$$D_{C_d}(u, v) = \begin{cases} D_C(u, v), & D_C(u, v) = d \\ 0, & D_C(u, v) \neq d \end{cases} \quad (10)$$

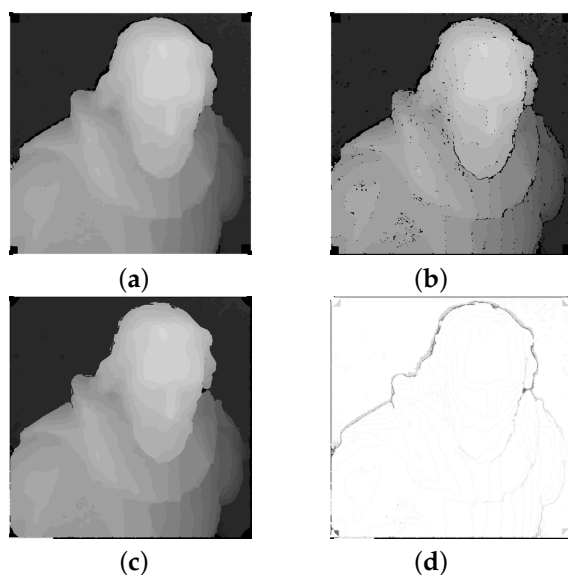
This image is a subject for the morphological closing operation [33], which stands for erosion, followed by dilation. Filtered disparity maps are combined back with preserving the presence of already associated pixels by processing the decomposed disparity maps from far to near. The resulting set of images is combined with the disparity map, used for the further borders generation step.

### 3.9. Holes Filling

After consistency check, merging and following filtering, the disparity map has a considerable amount of pixels without associated disparity value. In order to reduce the number of such pixels, we use a holes filling technique based on the neighborhood pixel information and color consistency. For a missing disparity pixel, the filling procedure is based on the median value of nearby values within a window of a certain size.

For the pixels, placed on the edges, such filling can lead to associating false values. To prevent it, we use values from a corresponding color light field view. Color pixels in the window are checked for their Euclidean distance from the reference pixel being below the threshold, based on which they are involved in the median value estimation.

This algorithm is performed iteratively, and the stopping criteria of this method are defined as the number of iterations and the number of non-empty pixels. To prevent jamming of the method on the same pixels after the third iteration, the window size is increased logarithmically and the threshold is relaxed accordingly. Figure 5 demonstrates the application of these steps to the initial disparity map.



**Figure 5.** Initial disparity post-processing steps: (a) initial disparity map, (b) per-layer disparity filtering, (c) holes filling, (d) difference between (a,c).

## 4. Final Disparity Map Estimation

This section describes how the previously estimated coarse disparity map is used for computations involving more light field views.

### 4.1. Generation of Borders

The initial disparity map serves for creation of computational limitation for disparity hypothesis range. It fulfills two purposes. First, the generation of matching cost from many



light field views is a time-consuming task. Limitation of the disparity search range by the bordering information reduces the running time of the matching procedure. Second, due to the ambiguities from the matching cost estimation, the wrong estimations and noise pixels can be present in the final disparity map. Bordering information prevents the appearance of these issues, which will be demonstrated later in Section 6.4.

$D_C$  is used for generation of boundaries for the further estimation. These boundaries will limit matching cost generation in the whole light field space. Two structures named high and low borders ( $D_H$  and  $D_L$  respectively) are generated by using the border threshold  $\lambda$  in such a manner:

$$D_H(u, v) = D_C(u, v) + \lambda; D_L(u, v) = D_C(u, v) - \lambda. \quad (11)$$

The values, which lie outside of predefined disparity range ( $D_H > d_{max}$ ,  $D_L < d_{min}$ ) are saturated accordingly. Invalid values from  $D_C$  are marked in the corresponding borders for re-computation on the whole disparity range  $T$ .

#### 4.2. Light Field Bordered Matching

The final disparity map can be estimated either from all images of the light field, or from the light field subset. Practically, it would be useful to utilize the views, lying on a cross from the reference. It can save the computational efforts on collecting the matching cost, whereas quality of the disparity would not be too much affected.

Matching cost for a pixel  $(u, v)$  for each possible disparity hypothesis  $d$ , lying in a range  $[D_L(u, v), D_H(u, v)]$  is found as a sum of compared values in all cross-lying views:

$$C_S(u, v, d) = \sum_{s=1}^{|s|} M(p(u, v, \hat{s}, \hat{t}, 0), p(u, v, s, \hat{t}, d)) + \sum_{t=1}^{|t|} C(p(u, v, \hat{s}, \hat{t}, 0), p(u, v, \hat{s}, t, d)), \quad (12)$$

where  $|s|$  and  $|t|$  are the numbers of views in the spatial light field dimensions, and  $M()$  is a comparison function.

The selection of this function depends on the origin and quality of the data. For synthetic data, usage of Euclidean distance is a fair choice, as it is done in our evaluation:

$$M(p_1, p_2) = \|p_1 - p_2\|. \quad (13)$$

However, for real-world scenarios, one would prefer Hamming distance-based comparison on Census-transformed images or more robust metrics like zero-normalized cross correlation [34].

Steps from Sections 3.5 and 3.6 are applied to  $C_S$  for obtaining final disparity map  $D_S$ .

#### 4.3. Sub-Pixel Refinement

Disparity map, computed with this method, contains only the values up to a pixel and can not be considered as accurate. As a post-processing step, we estimate the sub-pixel values of disparity pixels based on the matching cost. Usually, it is done by fitting a parabola to the neighboring cost values, associated with the disparity. However, this approach can produce a certain error, since, based on histogram analysis, the interpolated values are not equally distributed.

In this work, we use a technique called Symmetric-V interpolation, proposed by Haller et al. in [35].

For every pixel  $(u, v)$ , the values of interpolated image  $D_I$  are computed as:

$$D_I(u, v) = D_S(u, v) + \begin{cases} + \left( 0.5 - 0.25 \left( \frac{(M3-M1)^2}{(M2-M1)^2} + \frac{(M3-M1)}{(M2-M1)} \right) \right); M2 > M3 \\ - \left( 0.5 - 0.25 \left( \frac{(M2-M1)^2}{(M3-M1)^2} + \frac{(M2-M1)}{(M3-M1)} \right) \right); M2 \leq M3 \end{cases} \quad (14)$$

$$M1 = C_S(u, v, d), M2 = C_S(u, v, d - 1), M3 = C_S(u, v, d + 1)$$



## 5. Point Cloud Processing

The disparity map, described in the previous section, is further used as initialization for the point cloud, which will be optimized using all light field images. Such representation allows processing of all parts of the scene, presented in light field images, but are not visible in the reference view of the light field.

### 5.1. Point Cloud Conversion

For the point cloud generation, the disparity map  $D_I$  needs to be converted to the depth map, based on the common focal length of light field views  $f$  and the distance between two adjacent views on one axis  $b$ , which remains the same for all view pairs based on the light field parameterization:

$$D_Z(u, v) = \frac{fb}{D_I(u, v)}. \quad (15)$$

Depth values are used for getting the 3D points  $P$  as:

$$\begin{aligned} P &= [XYZ]; Z = D_Z(u, v) \\ X &= Z \frac{v}{f}; Y = Z \frac{u}{f} \end{aligned} \quad (16)$$

### 5.2. Additional Views Analysis

To include the information for scene points, which are not visible in the reference light field view, we define the reference views in the corners of the light field. The initial disparity map  $D_C$ , estimated in Section 3.6, is reprojected to the new reference view with the principles from Section 3.7. The reprojected image is then used for the generation of bordering information for the further estimation of final disparity map for the new reference view based on the cross-lying views, repeating steps from Section 4.

The point clouds are obtained by applying Equations (15) and (16) and transposed to the viewpoint of the original reference view. For multiple point cloud registration, we use a classical Iterative Closest Points (ICP) approach [36]. While ICP defines the needed transformation for all points, we remove the already preserved ones from the new point clouds by their projection to the original reference view plane and deleting the matching points. It allows for constructing the joined point cloud by just combining the point sets. For optimization purposes, the information about point origin is stored alongside with the point.

### 5.3. Nonlinear Optimization

Every point cloud is separately projected to various light field viewpoints. By that, we are trying to find value of  $Z$ , which minimizes the error between the projected and presented pixel in light field views:

$$\sum_{s=1}^{|s|} \|\hat{p}(u, v, s, \hat{t}, d) - p(u, v, s, \hat{t}, d)\| + \sum_{t=1}^{|t|} \|\hat{p}(u, v, \hat{s}, t, d) - p(u, v, \hat{s}, t, d)\|, \quad (17)$$

where  $\hat{p}$  is the projected pixel, estimated by the principles from Equation (16). Based on the point origin, it is optimized only on the frames, where the point is visible. For simplification, we define the possible configuration of viewpoints based on the cross-lying light field views.

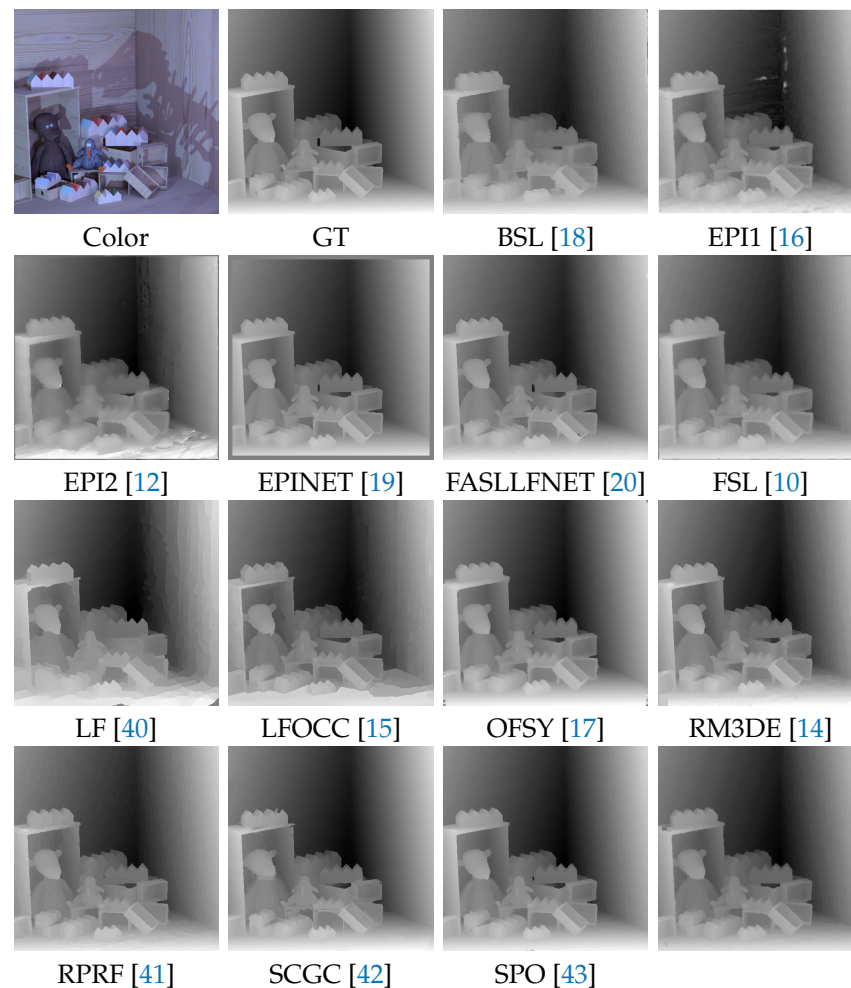
Output on the nonlinear optimization step contains some specific noise, which requires additional post-processing efforts. For reducing such noise, we found that the Combined Bilateral Filter (CBF), proposed by Wasenmüller et al. in [37], suits best. It composes the classical bilateral filter with joint bilateral filter, published in [38].

## 6. Results

### 6.1. Synthetic Dataset

Evaluation of our method is done with a four-dimensional Light Field Benchmark [39] on a synthetic dataset by Honauer et al. [11]. Twelve synthetic scenes are used for the comparison; each of them is represented by the  $9 \times 9$  light field, collected from 8-bit RGB images with  $512 \times 512$  pixel resolution. The images from the dataset used in the benchmark are divided into three categories: "training" for parameter adjustment and evaluation, "stratified" with difficult cases and "test" for "blind" verification. Disparity and depth maps are provided for "training" and "stratified" categories.

We provide a comparison of the proposed algorithm with the state-of-the-art methods, presented in Section 2: BSL [18], FASTLFNET [20], EPI1 [16], EPI2 [12], EPINET [19], FSL [10], LF [40], LFOCC [15], OFSY [17], RM3DE [14], RPRF [41], SCGC [42], and SPO [43]. Qualitative comparison for one of the light field images from "training" category is presented in Figure 6. Demonstration of the results on all scenes is presented on the web-site of 4D Light Field Benchmark [39].



**Figure 6.** Qualitative results for "dino" scene from 4D Light Field Benchmark [11].

### 6.2. Evaluation

The benchmark provides various metrics, on which algorithms can be evaluated. We provide results of the comparison on three metrics: the percentage of pixels where the absolute difference between the result and the ground truth is greater than the threshold, which is set to 7% (*BadPix*), mean square error over all pixels (*MSE*), and the maximum absolute disparity error of the best 25% of pixels (*Q25*). The results of the evaluation on these metrics are presented in Table 1. The benchmark provides various photo-consistency

metrics, which are not covered in this paper and can be found of the 4D Light Field Benchmark [39].

**Table 1.** Evaluation of different algorithms with general metrics on 4D Light Field Benchmark [11].

	<i>BadPix</i>		<i>MSE</i>		<i>Q25</i>	
	Median	Average	Median	Average	Median	Average
BSL [18]	13.41	12.74	5.43	7.28	0.92	1.01
EPI1 [16]	22.89	24.32	3.93	5.98	1.00	1.23
EPI2 [12]	22.94	22.65	5.72	8.24	0.71	0.81
EPINET [19]	<b>3.38</b>	<b>4.93</b>	<b>1.21</b>	2.48	0.34	<b>0.34</b>
FASTLFNET [20]	8.24	9.07	1.61	<b>2.46</b>	0.57	0.58
FSL [10]	11.92	12.95	3.97	6.64	0.85	0.95
LF [40]	16.15	16.19	7.96	9.13	0.58	0.61
LFOCC [15]	18.45	17.58	2.8	6.69	1.70	1.60
OFSY [17]	11.33	12.04	5.43	7.03	<b>0.32</b>	0.37
RM3DE [14]	7.99	10.22	1.46	3.92	0.73	0.72
RPRF [41]	9.89	10.02	3.76	5.68	0.66	0.64
SCGC [42]	10.21	14.3	3.94	6.58	1.04	1.09
SPO [43]	8.78	8.47	3.31	3.97	0.60	0.71
PSL (proposed)	11.61	12.79	2.78	5.14	0.93	0.89

### 6.3. Algorithm Settings

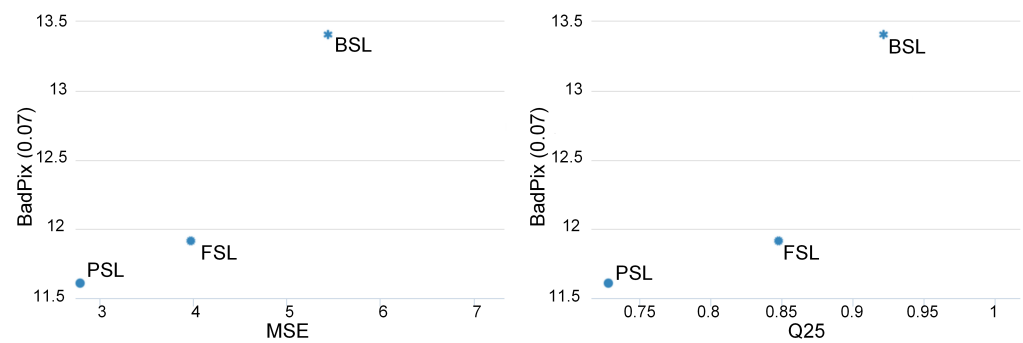
Every scene shares all of the algorithm parameters except the disparity range, which is adjusted according to the scene configuration file. The values are set empirically based on the optimal values of the evaluation on three metrics from Section 6.2.

Census transformation from Section 3.4 uses a window of size  $9 \times 7$ . SGM penalties  $P1$  and  $P2$  are set to 30 and 150 for the initial disparity map estimation from Section 3. Due to the different comparison formula, these penalties in Section 4 are set to 20 and 40. For both scenarios, the number of traversing directions for SGM equals 4.

Confidence threshold  $\varphi$  for the consistency check and merging in Equation (9) is set to 2. Holes filling algorithm in Section 3.9 uses 25 iterations as the stopping criteria of the optimization. Initial window size is set to 5 and initial threshold for the distance between color values of the pixels is set to 5. Border threshold  $\lambda$  from Equation equals 1. CBF from Section 5.3 uses standard deviation values of 0.5 and 2.5. Window size for median-based filters is set to 3.

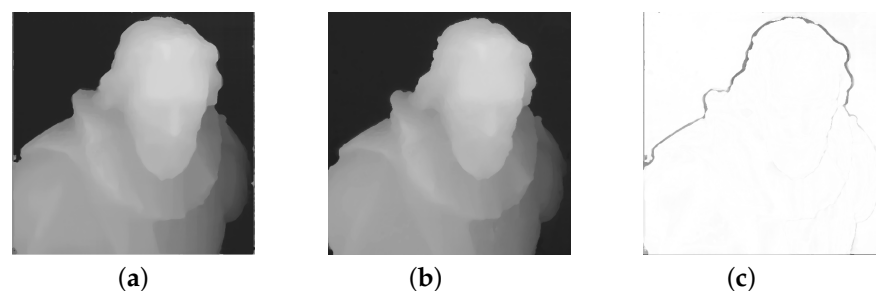
### 6.4. Discussion

Figure 7 shows how the result of the presented algorithm compares with its predecessors [10,18]. Overall, the quantitative results of our algorithm are consistent with the baseline. Values of three metrics for the benchmark were improved compared to the preceding algorithms, as demonstrated in Figure 7.



**Figure 7.** Comparison of the proposed algorithm (PSL) with its predecessors (BSL [18], FSL [10]) on the median of three metrics (lower is better), generated by [39].

Figure 8 shows the difference between predecessor [10] and the proposed algorithm. Two main changes can be observed. First, the smoothness of surfaces in the proposed algorithms is improved, as can be seen in Figure 8a,b. It can be observed on both final and refined disparity maps. This happens partially due to a change in the subpixel refinement algorithm from the parabola fitting to Symmetric-V. However, most of the smoothness is brought by a point cloud refinement step. It is not limited to the discrete matching cost values, unlike the interpolation step. In total, this corrects the step effect on disparity values, which was strongly observable in [10].



**Figure 8.** Effect of the per-layer disparity filtering and holes filling: (a) disparity map from FSL [10], (b) disparity map from proposed method, (c) difference between two disparity maps.

Second, per-layer disparity filtering together with holes filling provides significant changes for the final result. It makes the reconstruction look sharper and closer to the color projection. It can be seen in Figure 8c that the proposed algorithm provides different reconstruction on edges, which usually was a spot of ambiguities for the matching algorithm. It happens because pixels around edges were considered as a part of a neighborhood disparity layer due to the nature of the matching algorithm, which considers the interpolated color values of pixels among light field views. However, some wrong pixels can still "survive" the filtering, as it can be seen on the left side of the statue's head in Figure 8b. Potentially, it can be fixed by repeating the per-layer disparity filtering and holes filling several times.

Table 2 shows the quantitative difference of algorithm configurations with and without per-layer disparity filtering and holes filling, performed on a subset of images from the benchmark. It can be observed that filtering techniques not only affect the boundaries of the images, but also improve the accuracy of the algorithm.

Although the smoothness of surfaces is improved in our method, in terms of the benchmark metrics, our result is worse compared to deep learning methods. However, the advantage of our approach is that there is no need to provide training data. Such approach can be easily extended to the different configurations of cameras and to be used on various scenes as well, providing not perfect, but reasonable results.

**Table 2.** Average results on "training" subset of 4DLFB [39] for the configuration with and without per-layer disparity filtering and holes filling.

	<i>BadPix</i>	<i>MSE</i>	<i>Q25</i>
With	9.89	3.57	0.74
Without	10.23	5.72	0.72

Experiments with adaptive Census window did not show any significant improvement. In addition, in this approach, the dense window for Census transformation was used instead of a sparse one, unlike in our previous works. We found that the accuracy of initial disparity estimation is higher in such configuration.

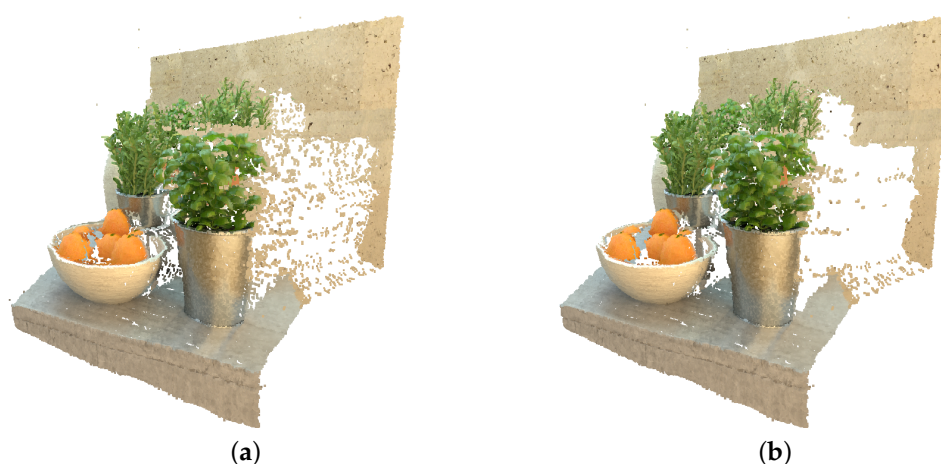
Usage of borders significantly reduces the number of sampled hypotheses. Due to the change of domain from disparity maps to point cloud, a new advantage of using the boundaries was observed. Previously, only the running time-related changes were noted; however, it turns out that image initialization also prevents the creation of noise pixels on the areas of big disparity values transition, as it can be observed in Figure 9.

A nonlinear optimization step requires a good initialization. In addition, such optimization on a full disparity range can unfortunately create additional false estimations. For that, the searching range is limited to 1.5 pixels around the initial value from the final disparity map.

One way of improvement of this step, as well as general generation of bordering information, is related to utilization of matching cost confidence measurements. Different thresholds can be used based on the accuracy for the specific pixel. A modern overview of methods for that is presented in [44].

Unlike previous approaches, the running time of the proposed one is significantly higher. The main reason for that is the nonlinear optimization. We are currently investigating ways of reducing the running time for these operations.

The presented method was performed on a central processing unit (CPU). It limits the running time, since no parallelism was exploited. One way of making this method faster is by bringing its parts to graphics processing units (GPU). Since most of the operations are performed separately per pixel, it can be done in parallel. More complicated steps, such as SGM, can be paralleled by the traversing directions.



**Figure 9.** Effect of disparity boundaries on the point cloud: (a) point cloud without borders, (b) point cloud with borders.

## 7. Conclusions

In this paper, we proposed an extension of the light field depth estimation method with the nonlinear point cloud refinement. Evaluation of our approach against state-of-the-art methods shows that results are comparable to the baseline result and improve its

predecessors. Further work will try to reformulate the algorithm with principles of deep learning and to improve the running time by utilizing parallelism of GPUs and downscaled initial structures.

**Author Contributions:** Conceptualization, Y.A. and D.S.; methodology, Y.A.; software, Y.A.; validation, Y.A.; formal analysis, Y.A.; investigation, Y.A.; resources, D.S.; data curation, Y.A.; writing—original draft preparation, Y.A.; writing—review and editing, J.R.R.; visualization, Y.A.; supervision, J.R.R. and D.S.; project administration, J.R.R.; funding acquisition, D.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially funded by the Federal Ministry of Education and Research (Germany) in the context of project DAKARA (13N14318).

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

CPU	Central Processing Unit
EPI	Epipolar–Plane Image
ICP	Iterative Closest Points
GPU	Graphic Processing Unit
SGM	Semi–Global Matching

### References

- Gershun, A. The light field. *J. Math. Phys.* **1939**, *18*, 51–151. [[CrossRef](#)]
- Adelson, E.H.; Bergen, J.R. *The Plenoptic Function and the Elements of Early Vision*; MIT Press: Cambridge, CA, USA, 1991; pp. 3–20.
- Levoy, M.; Hanrahan, P. Light field rendering. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 4–9 August 1996; pp. 31–42.
- Krolla, B.; Diebold, M.; Stricker, D. Light field from smartphone-based dual video. In Proceedings of European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 600–610.
- Wilburn, B.S.; Smulski, M.; Lee, H.H.K.; Horowitz, M.A. Light field video camera. In Proceedings of the SPIE Media Processors 2002, San Jose, CA, USA, 23–25 January 2002; pp. 29–36.
- Anisimov, Y.; Wasenmüller, O.; Stricker, D. A compact light field camera for real-time depth estimation. In Proceedings of International Conference on Computer Analysis of Images and Patterns, Salerno, Italy, 3–5 September 2019; pp. 52–63.
- Ng, R.; Levoy, M.; Brédif, M.; Duval, G.; Horowitz, M.; Hanrahan, P. Light field photography with a hand-held plenoptic camera. Ph.D. Thesis, Stanford University, California, CA, USA, April 2005.
- Kim, H.M.; Kim, M.S.; Lee, G.J.; Jang, H.J.; Song, Y.M. Miniaturized 3D Depth Sensing-Based Smartphone Light Field Camera. *Sensors* **2020**, *20*, 2129. [[CrossRef](#)] [[PubMed](#)]
- Isaksen, A.; McMillan, L.; Gortler, S.J. Dynamically reparameterized light fields. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, New Orleans, LA, USA, 23–28 July 2000; pp. 297–306.
- Anisimov, Y.; Wasenmüller, O.; Stricker, D. Rapid light field depth estimation with semi-global matching. In Proceedings of the 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj–Napoca, Romania, 5–7 September 2019; pp. 109–116.
- Honauer, K.; Johannsen, O.; Kondermann, D.; Goldluecke, B. A dataset and evaluation methodology for depth estimation on 4D light fields. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; pp. 19–34.
- Wanner, S.; Goldluecke, B. Globally consistent depth labeling of 4D light fields. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Rhode, USA, 16–21 June 2012; pp. 41–48.
- Bolles, R.C.; Baker, H.H.; Marimont, D.H. Epipolar-plane image analysis: An approach to determining structure from motion. *Int. J. Comput. Vis.* **1987**, *1*, 7–55. [[CrossRef](#)]
- Neri, A.; Carli, M.; Battisti, F. A multi-resolution approach to depth field estimation in dense image arrays. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec, Canada, 27–30 September 2015; pp. 3358–3362.
- Wang, T.C.; Efros, A.A.; Ramamoorthi, R. Occlusion-aware depth estimation using light-field cameras. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 2015; pp. 3487–3495.
- Johannsen, O.; Sulc, A.; Goldluecke, B. What sparse light field coding reveals about scene structure. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3262–3270.
- Strecke, M.; Alperovich, A.; Goldluecke, B. Accurate depth and normal maps from occlusion-aware focal stack symmetry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 26–27 July 2017; pp. 2814–2822.
- Anisimov, Y.; Stricker, D. Fast and efficient depth map estimation from light fields. In Proceedings of 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 337–346.



19. Shin, C.; Jeon, H.G.; Yoon, Y.; Kweon, I.S.; Kim, S.J. Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4748–4757.
20. Huang, Z.; Hu, X.; Xue, Z.; Xu, W.; Yue, T. Fast Light-Field Disparity Estimation With Multi-Disparity-Scale Cost Aggregation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; pp. 6320–6329.
21. Perra, C.; Murgia, F.; Giusto, D. An analysis of 3D point cloud reconstruction from light field images. In Proceedings of 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), Oulu, Finland, 12–15 December 2016; pp. 1–6.
22. Ferreira, R.; Gonçalves, N. Accurate and fast micro lenses depth maps from a 3D point cloud in light field cameras. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 1893–1898.
23. Farhood, H.; Perry, S.; Cheng, E.; Kim, J. Enhanced 3D point cloud from a light field image. *Remote. Sens.* **2020**, *12*, 1125. [[CrossRef](#)]
24. Yucer, K.; Kim, C.; Sorkine-Hornung, A.; Sorkine-Hornung, O. Depth from gradients in dense light fields for object reconstruction. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 249–257.
25. Liu, Y.; Dai, Q.; Xu, W. A point-cloud-based multiview stereo algorithm for free-viewpoint video. *IEEE Trans. Vis. Comput. Graph.* **2009**, *16*, 407–418.
26. Chen, R.; Han, S.; Xu, J.; Su, H. Point-based multi-view stereo network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1538–1547.
27. Yao, Y.; Luo, Z.; Li, S.; Fang, T.; Quan, L. Mvsnet: Depth inference for unstructured multi-view stereo. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 767–783.
28. Kim, C.; Zimmer, H.; Pritch, Y.; Sorkine-Hornung, A.; Gross, M.H. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.* **2013**, *32*, 73:1–73:12. [[CrossRef](#)]
29. Chen, C.C.; Chu, H.T. Similarity measurement between images. In Proceedings of the Computer Software and Applications Conference (COMPSAC), Edinburgh, UK, 26–28 July 2005; pp. 41–42.
30. Hirschmuller, H.; Scharstein, D. Evaluation of cost functions for stereo matching. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Minneapolis, MN, USA, 18–23 June 2007; pp. 1–8.
31. Zabih, R.; Woodfill, J. Non-parametric local transforms for computing visual correspondence. In Proceedings of the European Conference on Computer Vision, Stockholm, Sweden, 2–6 May 1994; pp. 151–158.
32. Hirschmuller, H. Accurate and efficient stereo processing by semi-global matching and mutual information. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 807–814.
33. Haralick, R.M.; Sternberg, S.R.; Zhuang, X. Image analysis using mathematical morphology. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *PAMI-9*, 532–550. [[CrossRef](#)] [[PubMed](#)]
34. Sabater, N.; Boisson, G.; Vandame, B.; Kerbiriou, P.; Babon, F.; Hog, M.; Gendrot, R.; Langlois, T.; Bureller, O.; Schubert, A.; et al. Dataset and pipeline for multi-view light-field video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 30–40.
35. Haller, I.; Pantilie, C.; Oniga, F.; Nedevschi, S. Real-time semi-global dense stereo solution with improved sub-pixel accuracy. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, La Jolla, CA, USA, 21–24 June 2010; pp. 369–376.
36. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *PAMI-9*, 698–700. [[CrossRef](#)] [[PubMed](#)]
37. Wasenmüller, O.; Bleser, G.; Stricker, D. Combined Bilateral Filter for Enhanced Real-time Upsampling of Depth Images. In Proceedings of the VISAPP (1), Berlin, Germany, 11–14 March 2015; pp. 5–12.
38. Kopf, J.; Cohen, M.F.; Lischinski, D.; Uyttendaele, M. Joint bilateral upsampling. *ACM Trans. Graph. (ToG)* **2007**, *26*, 96–es. [[CrossRef](#)]
39. 4D Light Field Benchmark. Available online: <https://lightfield-analysis.uni-konstanz.de> (accessed on 11 January 2022).
40. Jeon, H.G.; Park, J.; Choe, G.; Park, J.; Bok, Y.; Tai, Y.W.; So Kweon, I. Accurate depth map estimation from a lenslet light field camera. In Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, MA, USA, 7–12 June 2015; pp. 1547–1555.
41. Huang, C.T. Robust pseudo random fields for light-field stereo matching. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 11–19.
42. Si, L.; Wang, Q. Dense depth-map estimation and geometry inference from light fields via global optimization. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; pp. 83–98.
43. Zhang, S.; Sheng, H.; Li, C.; Zhang, J.; Xiong, Z. Robust depth estimation for light field via spinning parallelogram operator. *Comput. Vis. Image Underst.* **2016**, *145*, 148–159. [[CrossRef](#)]
44. Hu, X.; Mordohai, P. A Quantitative Evaluation of Confidence Measures for Stereo Vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2121–2133. [[PubMed](#)]