Fast and Efficient Depth Map Estimation from Light Fields

Yuriy Anisimov[1,2] and Didier Stricker[1,2]

[1]Department Augmented Vision, German Research Center for Artificial Intelligence (DFKI)
[2]Department of Computer Science, University of Kaiserslautern
http://av.dfki.de

Abstract

The paper presents an algorithm for depth map estimation from the light field images in relatively small amount of time, using only single thread on CPU. The proposed method improves existing principle of line fitting in 4-dimensional light field space. Line fitting is based on color values comparison using kernel density estimation. Our method utilizes result of Semi-Global Matching (SGM) with Census transform-based matching cost as a border initialization for line fitting. It provides a significant reduction of computations needed to find the best depth match. With the suggested evaluation metric we show that proposed method is applicable for efficient depth map estimation while preserving low computational time compared to others.

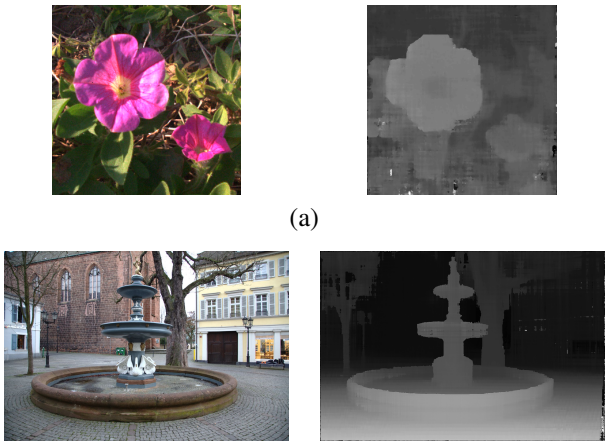


Figure 1. Results for real world scenes: (a) 4D light field, made from 7x7 images with resolution 328x328 from [16], and a result of proposed method, (b) image from 3D light field, made from 29 images with resolution 1409x938, and a result of proposed method

1. Introduction

The term "light field" originated in a monograph of Gershun, translated in [9], where it is formulated as the region of space, studied from the point of view of radiant energy transfer. Object of studies in this space is the rectilinear propagating ray with the radiant energy. This interpretation was then extended to the definition of plenoptic function, proposed by Adelson and Bergen in [2], which describes the information from any of the observing positions in space and time, and can be interpreted as parametrization of every possible location, viewing direction, wavelength and point in time; or, alternatively, in form where angles of direction are replaced by spatial coordinates of image plane.

As stated in [20], a light field can be determined as the radiance at a point in a given direction, and the plenoptic function may be reduced to 4-dimensional space. This leads us to the description of widely used representation of the ray in modern light fields — two-plane parametrization: a plane of spatial coordinates (u, v) stands for a 2-dimensional image in the light field, and angular plane (s, t) represents the viewpoint. Each light ray can be denoted as intersection of these two planes.

The proposed method considers a particular case of the light field, defined as a set of densely captured views of the scene. Shifting of the viewpoints for capturing can be performed by one or two moving directions, which for 2-dimensional images leads to 3-dimensional or 4-dimensional light field respectively.

Different ways of capturing light fields exist at present. One of the approaches is by using the so-called plenoptic camera [22] which consists of single image sensor with micro-lens array in front of it. Another way is a camera on moving stage, which allows shifting the device at equal length and capturing images of light field with same baseline. Similar capturing principle can be used with one- or two-dimensional arrays of multiple cameras.

Light field-based technologies and devices found their applications in industrial area (*e.g.* optical inspection), in

three-dimensional microscopy and in the cinema industry. All the light field applications require performing calculations related to scene depth value estimation. Large number of algorithms for such processing of light fields exist, and a short review of them is presented in Section 2. There is a trade-off between quality and runtime in the following algorithms: it takes a lot of time to process images with good results in the form of a depth map, or shorter amounts of processing time results in lower quality. Fast depth processing of light fields is not a trivial task, since big amount of data is involved and computations for the analysis methods are relatively complex.

The proposed method aims to compute a dense depth map of acceptable quality from a given 4D light field in relatively small amount of time with the possibility of further optimization. The approach is based on the realization of line fitting concept from [18] with additional Semi-Global Matching (SGM)-based[14] initialization. Results of SGM are good in terms of depth estimation for the whole image with low noise level, and the runtime of the algorithm is usually relatively small, but result for some image details (*e.g*. object borders, fine structures) is not very precise. On the other hand, line fitting principle gives good results in terms of detail preservation, however, the runtime is relatively high and there is a large level of noise in homogeneous regions. Without SGM information, line fitting goes through all the possible depth hypotheses, which is computationally intensive; with the SGM, runtime is reduced proportionally to determined pixel level. Section 3 provides detailed description of proposed method.

In Section 4, we show the output and evaluation of the algorithm, as well as faced limitations. Section 5 describes the conclusion and planned work, related to possible optimizations.

Our main contribution is the optimization of line fitting principle by utilizing results of SGM algorithm as bordering information with some optimizations. Also, in Section 4 we propose an evaluation metric related to number of correct pixels per second. With this metric it is showed that the performance of the algorithm is comparable to the state-of-the-art methods, while the time difference in most cases is significant.

2. Related work

2.1. Light field analysis

One of the first studies related to depth map estimation from 3D light fields was proposed by Bolles *et al*. [5]. It utilizes a structure derived from light field "slicing" and called "Epipolar-plane Image" (EPI). In the method detection of different features such as edges, peaks, and troughs is performed; results of the detection are used for line fitting in the EPI for structure estimation. Matoušek *et al*. [23] propose a dynamic programming solution for correspondences detection in the EPI, based on finding lines with similar intensities and minimization of a cost function.

Kim *et al*. [18] propose a method for precise scene reconstruction from dense sequence of high-resolution images using fine-to-coarse strategy with further propagation. It based on the EPI, with the line fitting function algorithm tests several depth hypotheses and picks the one which leads to the highest color density. Computed values, which are checked for confidence, used as bounds for depth calculation after EPI downsampling. The algorithm in [18] is computationally intensive and can be efficiently realized only using GPU. However, the principle of line fitting, based on kernel density estimation, formed the basis for our method. Instead of downsampling, borders are obtained using SGM results.

Jeon *et al*. [16] compute a cost volume from the shifting of sub-aperture images with gradient- and color-based similarity measurement. Refinement of the depth map, obtained by a winner-takes-all strategy, is performed using graph cuts [19]. In [31] and [32], Wanner and Goldluecke estimate a disparity map from light fields using EPI analysis with a structure tensor method, solving so-called "constrained labeling problem" with further variational regularization. Tao *et al*. [29] combines results of defocus and correspondence cue responses, calculated from the sheared EPI, to obtain the depth map. In [30] results of [29] are extended for the occlusion-handling case. Zhang *et al*. [35] propose a spinning parallelogram operator for depth estimation on a EPI.

Basha *et al*. [4] constructs dense volumetric representation of the three-dimensional space for estimating structure and motion using a multi-camera array. Because of the voxel representation of the scene, method can be considered as computationally-intensive. Neri *et al*. [25] presents multi-resolution method, based on local minimization of the maximum likelihood functional. The approach in [33] utilizes the patch-based local gradient information for depth calculation with further propagation. Navarro and Buades [24] propose a combination of two stereo non-dense methods, which in conjunction with interpolation gives a dense depth map.

2.2. Semi-global matching

Since SGM was invented in [14], several works related to it were published. SGM, which is used for border initialization, is based on pipeline, described in [11]. In proposed method, however, SGM uses cost aggregation method from the original paper [14], penalty parameter $P1$ is not excluded from the calculations. Sub-pixel interpolation is performed in the classic parabolic form. This approach was then improved by GPU implementation in [10].

CPU implementation based on 5x5 Census cost matching, proposed by Gehrig and Rabe in [8]. For reduction of

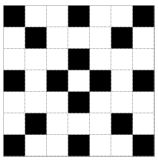


Figure 2. A pattern for sparse Census processing

memory consumption authors suggest to run SGM on subsampled images, on which one pixel is equal to the averaged value of four corresponding pixels in the original image. A pipeline for SGM with adaptive Census window is proposed by Loghman and Kim in [21]. An FPGA implementation of SGM is described in [3]. Hermann and Klette in [13] propose the iterative approach for SGM with invention of semi-global distance maps as a cost function alternative, which reduces the amount of data needed to process.

3. Algorithm description

In this section, we explain the proposed algorithm. 3.1-3.4 describe the steps for the line fitting borders initialization. These steps are performed for the first and last images of light field center row. Utilization of previous steps result in the form of a synthetic depth map described in 3.5. 3.6 gives an explanation for improving border quality with additional computations using the first and last images in the light field center column. Confidence measurement for depth map values from 3.4 with application to the synthetic depth map explained in 3.7. Additional edge filtration is provided in 3.8. Bordering information calculation is provided in subsection 3.9 and the line fitting approach is described in 3.10. Aggregation for a final depth map presented in 3.11.

3.1. Census transform-based matching cost

Census transform is a non-parametric transform, described in [34]. It is based on the comparison of radiance values between a pixel and its surroundings pixels within some window. To minimize the processing time while keeping capability of capturing whole image information, sparse Census window is used instead of densely filled one. The result of the transform is a bit string obtained as

$$I_c(u,v) = \bigotimes_{[i,j]\in D} \xi(I(u,v),\, I(u+i, v+j)), \qquad (1)$$

where I_c is Census-transformed image, I stands for the grayscaled image, D is a set with coordinates of window elements used for transform, $\otimes$ stands for bitwise concatenation. Pixels relation is given by:

$$\xi(p_1,p_2) = \begin{cases} 0, & p_1 \leqslant p_2 \\ 1, & p_1 > p_2 \end{cases}. \qquad (2)$$

In our approach, pixel sampling for cost function is performed in both possible shifting directions (forward and backward); it leads to a possibility of SGM calculations for such images, in which the background and foreground move to the different directions. Cost calculation between pixels in two Census-transformed images is determined as a Hamming distance [12] between pixels in both images

$$C(u,v,d) = HD(I1_c(u,v), I2_c(u+d,v)), \qquad (3)$$

where C – structure with cost calculations result, $I1_c$ and $I2_c$ are first and second Census-transformed images, d – depth hypothesis, equal to pixel shift movement, which lies between maximum hypothesis level for moving in direction $d1$ and $d2$ ($d1_{max}$ and $d2_{max}$ respectively). HD stands for Hamming distance, which is calculated between two vectors x_i and x_j with same size n, as a quantity of elements with different values ($\oplus$ denotes exclusive disjunction)

$$HD(x_i,x_j) = \sum_{k=1}^{n} x_{ik} \oplus x_{jk}. \qquad (4)$$

3.2. Semi-global matching

Costs are aggregated using the principle from original SGM paper [16]. For each pixel $p = (u,v)$ and depth hypothesis d, after traversing in direction r (determined as 2-dimensional vector with coordinate of pixel traversing $r = \{\Delta u, \Delta v\}$), aggregated cost L_r is

$$\begin{aligned} L_r(p,d) &= C(p,d) + \\ &min\,(L_r(p-r,d), \\ &L_r(p-r,d-1) + P1, \\ &L_r(p-r,d+1) + P1, \\ &\min_t L_r(p-r,t) + P2), \end{aligned} \qquad (5)$$

where t lies between $-d1_{max}$ and $d2_{max}$. Aggregated costs are summarized through all traversing directions:

$$C_s(p,d) = \sum_r L_r(p,d). \qquad (6)$$

From the cost summary, initial depth value is calculated using the winner-takes-all principle as

$$D_{init}(p) = \arg\min_d C_s(p,d). \qquad (7)$$

3.3. Interpolation

Refinement of the initial depth map is performed by parabolic interpolation of cost summary values

$$\begin{aligned} D_{sub}(p) &= D_{init}(p) + \\ &\frac{C_s(p,d-1) - C_s(p,d+1)}{2*(2*C_s(p,d) - C_s(p,d-1) - C_s(p,d+1))}. \end{aligned} \qquad (8)$$

As a result of interpolation, depth values with sub-pixel accuracy are obtained. It gives a smoother outlook for the depth map without quality loss and allows us to calculate borders in 3.9 more accurately.

3.4. Left-right consistency check

For occlusion filtering, steps 3.1-3.3 are calculated relative to left (left-right matching) and right (right-left matching) images, resulting in two sets of two depth maps $\{D_{Linit}, D_{Lsub}\}$ and $\{D_{Rinit}, D_{Rsub}\}$. Reliability of depth value in two depth maps for pixel p is estimated through confidence measure

$$CMT_{LR}(p) = \begin{cases} 1, & |D_L(p) - D_R(p)| < \varphi \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

where φ stands for confidence threshold.

3.5. Synthetic depth map

Previously obtained depth maps are used for construction of a new synthetic depth map, called D_{syn}. For that, interpolated depth maps need to be shifted to fit the position of the central view of the light field. Several methods can be used here for the translation distance determination: with known baseline and camera parameters translation can be calculated directly, in other cases information of phase correlation between images might be applicable. For shifted depth maps D_{Lsub_S} and D_{Rsub_S}, per-pixel averaging of the depth values is performed:

$$D_{syn}(p) = (D_{Lsub_S}(p) + D_{Rsub_S}(p))/2. \quad (10)$$

3.6. Top-bottom Semi-global matching

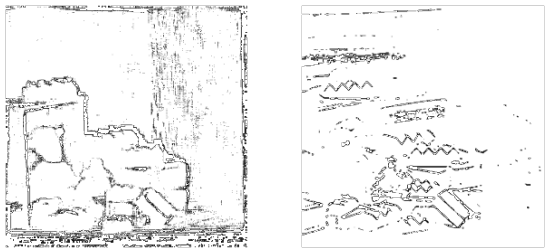
For improving the quality of the result, similar calculations can be done for the first and last images in light field center column. (3) becomes

$$C(u, v, d) = HD(I1_c(u, v),\ I2_c(u, v + d)), \quad (11)$$

and steps, described in 3.2-3.4 are repeated with respect to changed moving direction (from top to bottom).

As a result, we obtain sets of depth maps $\{D_{Tinit}, D_{Tsub}\}$,$\{D_{Binit}, D_{Bsub}\}$, and confidence measurement CMT_{TB}. These depth maps can be used together with $\{D_{LsubS}, D_{RsubS}\}$ for the construction of a more accurate synthetic depth map. Shifting on depth maps (from 3.5) need to be performed, and for the set of shifted depth maps $\{D_{LsubS}, D_{RsubS}, D_{TsubS}, D_{BsubS}\}$, (10) becomes

$$\begin{aligned} &D_{syn}(p) = \\ &(D_{Lsub_S}(p) + D_{Rsub_S}(p) + \\ &(D_{Tsub_S}(p) + D_{Bsub_S}(p))/4. \end{aligned} \quad (12)$$



(a) (b)

Figure 3. Intermediate image structures of the algorithm (with top-bottom SGM) for "dino" dataset from [15]: (a) CMT_{syn} in 3.7, (b) E_{syn} in 3.8

3.7. Confidence measurement

Confidence measurement, previously mentioned in 3.4, is used for definition of a similar one for synthetic depth map CMT_{syn}. For left-right only matching, it is equal to CMT_{syn}; with included top-bottom SGM result is calculates as

$$CMT_{syn}(p) = \begin{cases} 1, & CMT_{LR}(p) = CMT_{TB}(p) \\ 0, & \text{otherwise} \end{cases}. \quad (13)$$

3.8. Edges exclusion

As mentioned before, SGM does not provide precise results on boundaries of the objects. This motivates us to exclude the bordering information for edges, obtained from center view of light field. For this, edges are calculated using the Sobel operator [27] and smoothed using median filter; corresponding points are then stored in E_{syn} structure. According to the experiments, this strategy works better in terms of precision of the final result than a scan using borders on edges from the SGM result. However, amount of data needed to be processed by the line fitting algorithm increases in this case, which affects the runtime.

3.9. Bordering information

In order to calculate borders for line fitting concept, an intermediate structure D_{brd} is created. Pixels, considered as unreliable in CMT_{syn} and edges from E_{syn}, are marked in this structure to be calculated without border values, so full scan for every possible value will be done. For other values, normalization to line fitting coordinates is performed. Here, we introduce two parameters for the line fitting: depth window DW and depth step DS. They determine, respectively, the range of search for the line to be fitted in light field and slope of the minimal depth hypothesis. Number of depth hypotheses for line fitting calculated as $N = DW/DS$. From this, coefficient for normalization of bordering information is computed as

$$k_{brd} = N/(d1_{max} + d2_{max}). \quad (14)$$

With this data, values in D_{brd} can be estimated through

$$D_{brd}(p) = (D_{syn}(p) + dI_{max})k_{brd} \tag{15}$$

if $CMT_{syn}(p) = 1$ and $E_{syn}(p) = 0$; and 0 otherwise. Using the D_{brd} structure, low and high bordering values for each pixel are calculated with the algorithm 1 (λ is a border penalty parameter) and stored respectively in B_L and B_H structures.

3.10. Line fitting

As mentioned in 2.1, line fitting principle origins from [18]. We use the density estimation function to calculate depth score in areas bordered by previous steps. As a pivot image for calculations the center image of the light field is selected. It is denoted as $(\hat{s} = \lceil n/2 \rceil, \hat{t} = \lceil m/2 \rceil)$ in coordinate system of the light field, where n and m – number of horizontal and vertical views in light field respectively. For each possible hypothesis of light field pixel of coordinates (u, v) with respect to the pivot image density value is calculated as

$$\begin{aligned} S(u,v,d) = &\sum_{s=1}^{n}\sum_{t=1}^{m} \\ &K(L(u+(\hat{s}-s)d, v+(\hat{t}-t)d, s, t) - \\ &L(u,v,\hat{s},\hat{t})), \end{aligned} \tag{16}$$

where L is 4-dimensional light field, s and t – horizontal and vertical position of image in light field.

Radiance similarity is verified using a optimized Parzen estimation method [6] with a Epanechnikov kernel [7]. For a given vector x:

$$K(x) = \begin{cases} 1-l, & l \leqslant 1 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$$l = \sum_{i=1}^{c} x_i^2 \upsilon \tag{18}$$

$$\upsilon = 1/h^2, \tag{19}$$

where h stands for the bandwidth parameter, c — number of elements in vector.

Unlike the original approach [18], we use a simplified calculation of the kernel, *e.g.* we avoid square root calculations, which reduces the computational time.

3.11. Final depth map

The final depth result is determined for the center light field view according to the highest value of the density sampling and saved in D_{final}:

$$D_{final}(u,v) = \arg\max_{d} S(u,v). \tag{20}$$

Algorithm 1 Calculation of borders for line fitting

```
for p = 1 to pixels do
    if D_brd(p) = 0 then
        B_L(p) = 0
        B_H(p) = N
    else
        if D_brd(p) − λ ⩾ 0 then
            B_L(p) = D_brd(p) − λ
        else
            B_L(p) = 0
        end if
        if D_brd(p) + λ ⩽ N then
            B_H(p) = D_brd(p) + λ
        else
            B_H(p) = N
        end if
    end if
end for
```

Median filter is applied to final result to remove noise.

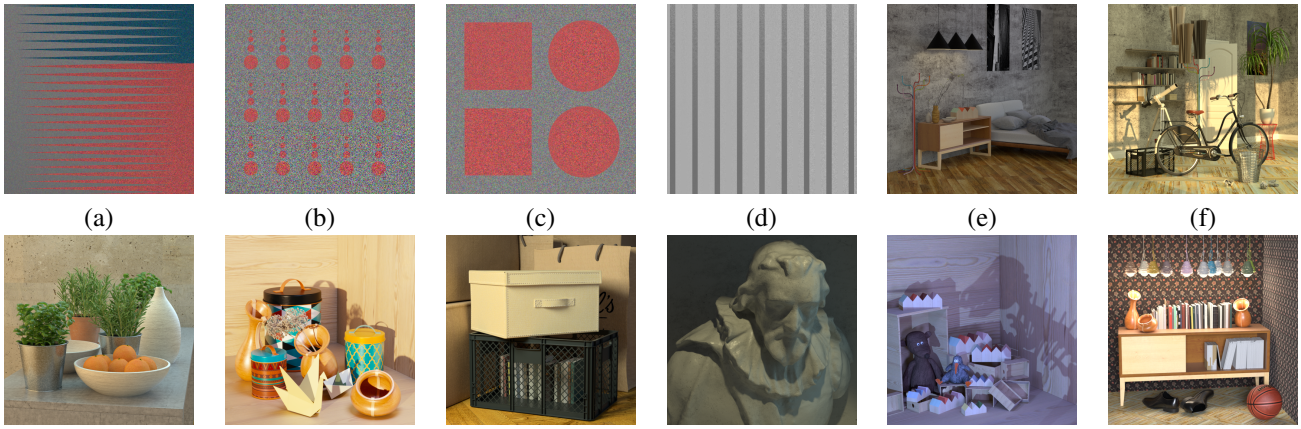
For memory efficiency purposes, values of S in (16) are not stored during the processing time, after highest score estimation they are overwritten by the processing information for next pixels.

4. Experiments

In this section we provide the comparison of the proposed method with presented in Section 2 state-of-the-art algorithms [35, 28, 25, 26, 17, 31, 16, 30]. In tables and figures these algorithms are presented under acronyms SPO, OFSY, RM3DE, SC_GC, EPI1, EPI2, LF and LF_OCC respectively. Evaluation is carried out by the 4D Light Field Benchmark [1] [15].

4.1. Datasets

We use the light field images, provided by Honauer *et al.* [15] through 4D Light Field Benchmark. 12 synthetic scenes are provided for the main evaluation; each scene is represented by the 9x9 light field, composed from 8-bit RGB images with resolution of 512x512 pixels. Datasets are grouped in three categories: "training" for evaluation and parameters adjustment, "stratified" with special challenging cases, and "test" for "blind" verification. Camera settings and disparity ranges provided for every light field, high resolution disparity and depth maps are provided only for "training" and "stratified" datasets. In this section we present image result comparison for "dino" and "cotton" datasets; results for other datasets can be found at [1] under the *BSL* acronym, and in supplementary materials.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Figure 4. Center images of light fields from [15]: (a) "backgammon", (b) "dots", (c) "pyramids", (d) "stripes", (e) "bedroom", (f) "bicycle", (g) "herbs", (h) "origami", (i) "boxes", (j) "cotton", (k) "dino", (l) "sideboard"

4.2. Metrics

Benchmark provides several metrics for result evaluation. Together with some general measurements, like Mean Squared Error, algorithms can be evaluated in terms of some photorealistic terms, *e.g.* surface smoothness. Main criteria for evaluation of our method is the estimation of the percentage of errors in algorithm result, formulated as the $BadPix$ metric in mentioned benchmark, and the running time of the algorithm. $BadPix$ stands for the percentage of pixels in which absolute difference of result and ground truth bigger than T, where T set to 0.07. Corresponding formulas and description can be found in [15]. For purposes of interpreting our result in terms of stated in Section 1 contributions, we propose a metric M. This metric stands for percentage of correctly computed pixels per second, formulated as

$$M = \frac{100\% - BadPix}{Runtime}\left(\frac{\%}{sec.}\right), \tag{21}$$

We provide an average and median result out of metrics mentioned above in tables, result for all datasets separately, together with other metrics, can be found at [1].

4.3. Algorithm settings

Mentioned in Section 3 parameters were adjusted for the best evaluation result in $BadPix$ metric and the runtime. Algorithm parameters stayed fixed independently of scene parameters except the disparity range. Penalty parameters for the SGM $P1$ and $P2$ were set to 21 and 45. Number of possible disparities (pixel shifting in both possible directions) is adjusted accordingly to the data, provided in configuration files for each of the scene. For the Census transform, used in SGM, different patterns of the aggregation window have been evaluated, for the experiments we use 7x7 pattern (Fig. 2). Another options for sparse Census window are listed in [21].

Edge exclusion was not performed, and SGM is calculated for the left-right image pair only. These adjustments related to runtime optimization of the algorithm.

The range for line fitting is set corresponding to disparity ranges for each of the datasets, and sampling line step is set as $(1/(N-1))\tau$, where N corresponds to number of images in one light field dimension ($N = 9$ in our case), and τ stands for step coefficient which is set to $1/7$. The kernel size of the median filter for final filtration is 3. The confidence threshold φ in (9) has been set to 3 and border penalty λ in Alg.1 – to 2. Bandwidth parameter h in (19) is set to 0.02.

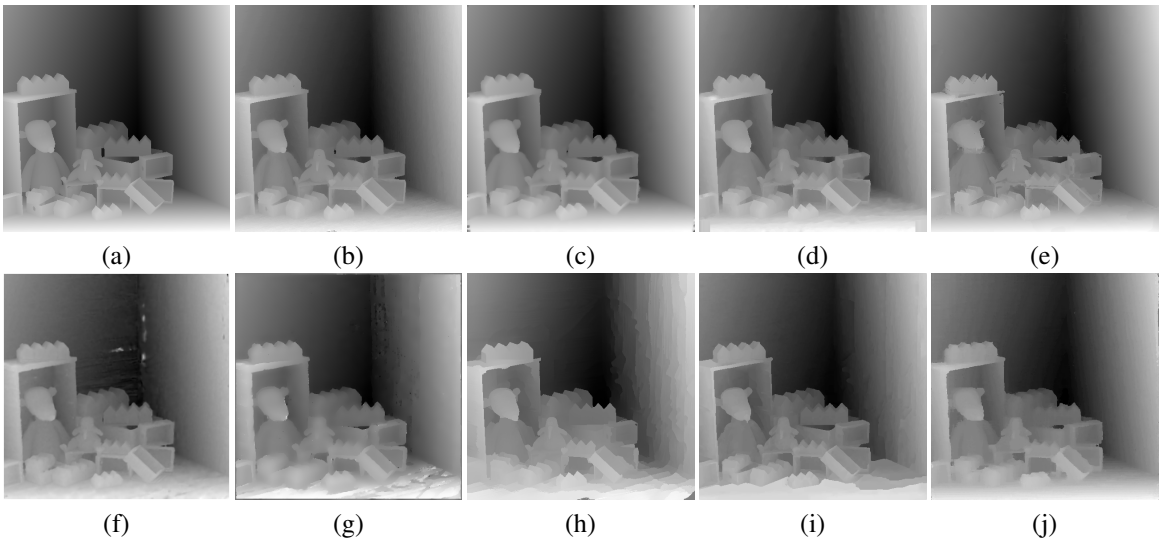
4.4. Visualization

Fig. 5 and 6 illustrate the depth estimation result on the "dino" and "cotton" light field datasets from benchmark. For "dino" dataset result faces some problems with sharpness of some fine structures; also, "step" effect on the wall can be noticed. Borders of some object are not accurate, and the same issue appears in the result of "cotton" dataset (noise on the head contour). In general, result of the proposed algorithm processing, compare to others, for these two datasets is on the average in terms of subjected level of quality.

4.5. Results

Table 1 shows the comparison of algorithm by *BadPix* metric, described in subsection 4.2. Result of the proposed method is on the average position compare to others.

Table 2 represents the runtime of algorithms, reported by authors of evaluated methods in 4D Light Field Benchmark. According to this result, runtime of our method is



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 5. Results for "dino" dataset from [15]. (a) Ground truth, (b) SPO [35], OFSY [28], (d) RM3DE [25], (e) SC_GC [26], (f) EPI1 [17], (g) EPI2 [31], (h) LF [16], (i) LF_OCC [30], (j) proposed method

	SPO [35]	OFSY [28]	RM3DE [25]
Median	8.779	11.329	**7.992**
Average	**8.466**	12.036	10.216
	SC_GC [26]	EPI1 [17]	EPI2 [31]
Median	10.206	22.891	22.942
Average	14.299	24.324	22.651
	LF [16]	LF_OCC [30]	proposed
Median	16.146	18.451	13.409
Average	16.193	17.579	12.743

Table 1. The percentage of pixels in which absolute difference of result and ground truth larger than threshold T ($BadPix$) on 4D Light Field Benchmark [15]. Here $T = 0.07$

	SPO [35]	OFSY [28]	RM3DE [25]
Median	2111.500	198.299	45.149
Average	2115.417	200.282	47.434
	SC_GC [26]	EPI1 [17]	EPI2 [31]
Median	2052.190	85.045	8.789
Average	2056.344	88.194	8.406
	LF [16]	LF_OCC [30]	proposed
Median	994.311	10614.54	**5.149**
Average	1009.756	10508.47	**5.962**

Table 2. Runtime in seconds on 4D Light Field Benchmark [15]

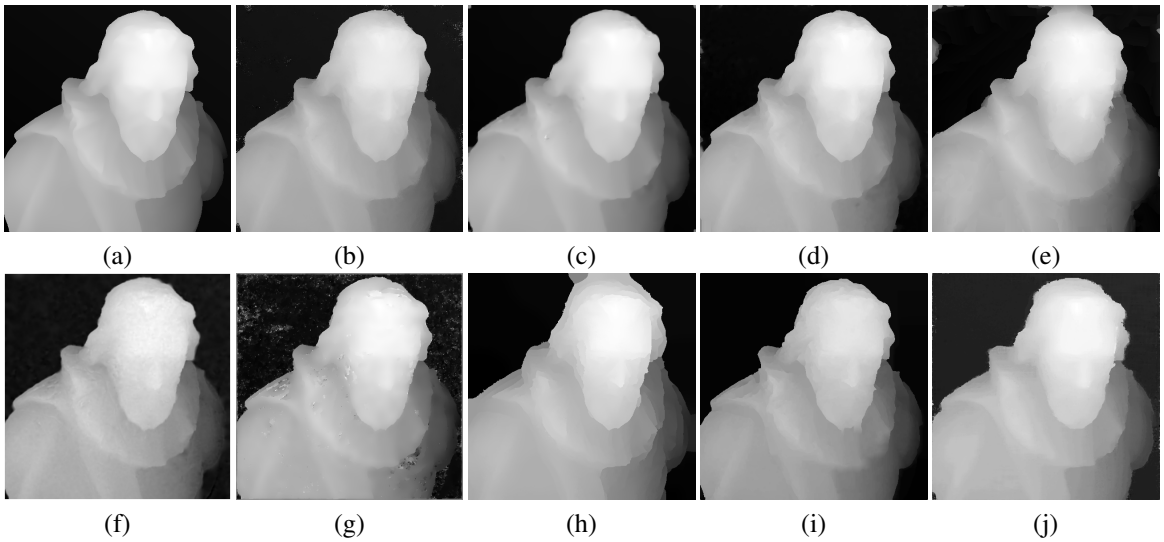
better than in most of the state-of-the-art algorithms. In some scenes, SGM results were eliminated for a large number of pixels, and line fitting without bordering information for these pixels affected the runtime (datasets "bicycle", "herbs", "boxes" and "sideboard"). This occurs because of the amount of fine structures in the scene. Application of a smaller window for Census transform in SGM seems to be a solution in terms of accuracy of the borders for the fine structures; however, it reduces the quality of the whole image, hence results of these tests are not provided. Some of the algorithms in the comparison were run on GPU architecture in contrast to our single thread algorithm. Parallelization for our method is available because of the nature of the algorithm. Parallel versions of SGM are covered in different papers [3, 10], and for line fitting scan can be performed line-by-line in parallel, since no result of calculations on different lines is used.

Proposed algorithm outperforms the majority of algorithms in the benchmark in terms of percentage of correctly calculated pixels in the image per second. Results of the comparison with the proposed in (21) metric are presented in Table 3. For 12 images benchmark our result is the best in 10 cases; average and median value of the metric is the top-of-the-line.

4.6. Real world scenes

Fig. 1 shows results for two real world scenes. First scene is represented as a 3D light field and contains 29 images with resolution of 1409x938 pixels. It was acquired using the moving stage with a camera on it. The baseline between each image is about 5 mm. For this dataset SGM penalty parameters $P1$ and $P2$ were set to 71 and 105 respectively.

Second scene provided by Jeon *et al*. in [16] and obtained by lenslet-based light field camera. The 4D light field contains of 7x7 images with resolution of 328x328 pixels. Algorithm uses same parameters from 4.3, except step co-



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 6. Results for "cotton" dataset from [15]. (a) Ground truth, (b) SPO [35], OFSY [28], (d) RM3DE [25], (e) SC_GC [26], (f) EPI1 [17], (g) EPI2 [31], (h) LF [16], (i) LF_OCC [30], (j) proposed method

	SPO [35]	OFSY [28]	RM3DE [25]
Median	0.044	0.458	1.961
Average	0.043	0.465	1.952
	SC_GC [26]	EPI1 [17]	EPI2 [31]
Median	0.043	0.911	9.054
Average	0.042	0.867	9.310
	LF [16]	LF_OCC [30]	proposed
Median	0.082	0.008	**18.392**
Average	0.083	0.009	**22.247**

Table 3. Percentage of correctly calculated pixels per second (21) on 4D Light Field Benchmark [15]

efficient τ, which was set to $1/5$.

Visually the result is fine; however, some challenges, *e.g.* small and fine structures, are noticeable. Processing of these scenes took 4.34 sec. and 411 ms respectively. We do not provide the comparison with other metrics from 4.2, since no ground truth is available for these images.

4.7. Environment

Execution of the proposed method was performed on CPU E3-1245 V2 @ 3.40 GHz, forced to work in a single thread. The proposed algorithm is implemented in C and compiled in Arch Linux using GCC v.7.1.1 with /O3 option.

4.8. Limitations

During the experiments, several disadvantages of our approach have surfaced. Estimated depth maps are noisy in discontinuities area, for some images a "step" effect of depth change is preserved (datasets "bedroom", "boxes", "dino"); in the regions with random noise pattern, acceptable depth values are not calculated (dataset "dots"). Algorithm shows the average (and for a part of cases — relatively bad) result in terms of evaluation of the proposed in [15] photorealistic metrics. Quality-related optimizations need to be done for avoiding these problems.

Also, subjective sharpness level of some objects is related to the selected configuration for the mentioned in 3.5 D_{syn} structure. It drops if we use only left-right SGM result. Conjunction with top-bottom SGM (13) increases the object sharpness, but it requires more time for SGM calculations. Line fitting also suffers from that, since more pixels are marked as unreliable with further full scan for them.

5. Conclusion

In this paper, we presented an algorithm for depth estimation from light field images, which combines stereo matching and line fitting approaches. We verified our algorithm with the different metrics, and the result of evaluation showed us that the proposed method is achieving a comparable to state-of-the-art depth map result. The method shows one of the best runtime and outperforms most of the state-of-the-art algorithms with the proposed metric (21). For future work, we plan to add runtime-related modifications (parallelization, SIMD-instructions), use an adaptive Census window for the special cases and also involve the gradient information for matching and additional confidence measurements.

Ackowledgments

This work has been partially funded by the BMBF project DAKARA (13N14318). The authors are grateful to Vladislav Golyanik, Kiran Varanasi and Jonathan Wray for the provided help.

References

[1] 4d light field benchmark. http://hci-lightfield.iwr.uni-heidelberg.de. Accessed: 18.07.2017. 5, 6

[2] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. 1991. 1

[3] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch. Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation. In *Embedded Computer Systems (SAMOS), 2010 International Conference on*, pages 93–101. IEEE, 2010. 3, 7

[4] T. Basha, S. A., A. Hornung, and W. Matusik. Structure and motion from scene registration. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1426–1433. IEEE, 2012. 2

[5] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International journal of computer vision*, 1(1):7–55, 1987. 2

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012. 5

[7] V. A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969. 5

[8] S. K. Gehrig and C. Rabe. Real-time semi-global matching on the cpu. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 85–92. IEEE, 2010. 2

[9] A. Gershun. The light field. *Studies in Applied Mathematics*, 18(1-4):51–151, 1939. 1

[10] I. Haller and S. Nedevschi. Gpu optimization of the sgm stereo algorithm. In *Intelligent Computer Communication and Processing (ICCP), 2010 IEEE International Conference on*, pages 197–202. IEEE, 2010. 2, 7

[11] I. Haller, C. Pantilie, F. Oniga, and S. Nedevschi. Real-time semi-global dense stereo solution with improved sub-pixel accuracy. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 369–376. IEEE, 2010. 2

[12] R. W. Hamming. Error detecting and error correcting codes. *Bell Labs Technical Journal*, 29(2):147–160, 1950. 3

[13] S. Hermann and R. Klette. Iterative semi-global matching for robust driver assistance systems. In *Asian Conference on Computer Vision*, pages 465–478. Springer, 2012. 3

[14] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005. 2

[15] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision*. Springer, 2016. 4, 5, 6, 7, 8

[16] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. So Kweon. Accurate depth map estimation from a lenslet light field camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1547–1555, 2015. 1, 2, 5, 7, 8

[17] O. Johannsen, A. Sulc, and B. Goldluecke. What sparse light field coding reveals about scene structure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3270, 2016. 5, 7, 8

[18] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. H. Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73–1, 2013. 2, 5

[19] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *Computer Vision-ECCV 2002*, pages 8–40, 2002. 2

[20] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996. 1

[21] M. Loghman and J. Kim. Sgm-based dense disparity estimation using adaptive census transform. In *Connected Vehicles and Expo (ICCVE), 2013 International Conference on*, pages 592–597. IEEE, 2013. 3, 6

[22] A. Lumsdaine and T. Georgiev. The focused plenoptic camera. In *Computational Photography (ICCP), 2009 IEEE International Conference on*, pages 1–8. IEEE, 2009. 1

[23] M. Matoušek, T. Werner, and V. Hlavác. Accurate correspondences from epipolar plane images. In *Proc. Computer Vision Winter Workshop*, pages 181–189. Citeseer, 2001. 2

[24] J. Navarro and A. Buades. Robust and dense depth estimation for light field images. *IEEE Transactions on Image Processing*, 26(4):1873, 2017. 2

[25] A. Neri, M. Carli, and F. Battisti. A multi-resolution approach to depth field estimation in dense image arrays. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3358–3362. IEEE, 2015. 2, 5, 7, 8

[26] L. Si and Q. Wang. Dense depth-map estimation and geometry inference from light fields via global optimization. In *Asian Conference on Computer Vision*, pages 83–98. Springer, 2016. 5, 7, 8

[27] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing (1968). *a talk at the Stanford Artificial Intelligence Project*, 1968. 4

[28] M. Strecke, A. Alperovich, and B. Goldluecke. Accurate depth and normal maps from occlusion-aware focal stack symmetry. 2017. 5, 7, 8

[29] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 673–680, 2013. 2

[30] T.-C. Wang, A. Efros, and R. Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3487–3495, 2015. 2, 5, 7, 8

[31] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4d light fields. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 41–48. IEEE, 2012. 2, 5, 7, 8

[32] S. Wanner and B. Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):606–619, 2014. 2

[33] K. Yucer, C. Kim, A. Sorkine-Hornung, and O. Sorkine-Hornung. Depth from gradients in dense light fields for object reconstruction. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 249–257. IEEE, 2016. 2

[34] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, pages 151–158. Springer, 1994. 3

[35] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong. Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding*, 145:148–159, 2016. 2, 5, 7, 8