Fast and accurate micro lenses depth maps for multi-focus light field cameras

Rodrigo Ferreira and Nuno Goncalves

Institute of Systems and Robotics - University of Coimbra
Coimbra, Portugal
{rodrigotoste@isr.uc.pt, nunogon@deec.uc.pt}

Abstract. Light field cameras capture a scene's multi-directional light field with one image, allowing the estimation of depth. In this paper, we introduce a fully automatic fast method for depth estimation from a single plenoptic image running a RANSAC-like algorithm for feature matching. The novelty about our approach is the global method to back project correspondences found using photometric similarity to obtain a 3D virtual point cloud. We then use lenses with different focal-lengths in a multiple depth map refining phase and their reprojection to the image plane, generating an accurate depth map per micro lens. Tests with simulations and real images are presented and show a good trade-off between computation time and accuracy of the method presented. Our method achieves an accuracy similar to the state-of-the-art in considerable less time (speedups of around 3 times).

1 Introduction

Plenoptic or light field cameras (PLF) are cameras that acquire the plenoptic function, that is to say that they know, for each pixel, the amount of light traveling in all directions. These cameras have received a lot of interest in the few last years since they inherently allow for multiple view geometry. Although formalized earlier (about 100 years ago), PLF cameras were commercially built only in the last decade. These cameras are built by placing a micro-lens array behind the major optical lens of the system. This construction allows for the formation of an array of smaller images that compose the 4D light field and by easily sampling it. It is then straightforward to estimate the scene's depth due to the redundancy created by the same point being imaged several times.

The concept behind plenoptic cameras was first addressed in 1908 by Lippmann [7] where he suggests the placement of an array of lenses between the camera's main lens and the film. This approach allows the camera to capture the light field of a scene. The concept was later refined by Ives [4] in 1930 but, due to the lack of computational power or existence of digital image sensors, little could be done to extract information from the light field. Now with digital image sensors, this technology has several possible applications such as robotics, face recognition, photography and filmography, augmented reality, depth reconstruction, industrial inspection and more.

Concerning depth estimation from plenoptic images we are able to achieve the scene depth with only one raw image, which is also essential for image rendering.

In 2004 Dansearau and Bruton [2] proposed a method for depth estimation using 2D gradient operations. They were able to define the light field direction and thus the depth of the corresponding elements within the light field. The areas where the depth could not be estimated were filled by applying region growing. Since plenoptic cameras are not immune to spatial aliasing, which can result on depth estimation errors, in 2009 Bishop and Favaro [1] applied a different approach to compensate the present aliasing, allowing them to recover the depth map from the multiple views provided by the 4D light field.

Wanner and Goldluecke [14] presented in 2012 a technique for depth estimations for 4D light fields, using dominant directions on epipolar plane images. By assuming that the 4D light field can be sliced onto 2D dimensions they started to locally estimate the depth of the epipolar plane images and then labeled the local estimations, integrating them on the global depth maps by imposing spatial constraints. Recently, Fleischmann and Koch [3] approach the depth estimation paradigm with disparity between neighbor lenses. Their method requires a very dense sampling of the light field. The micro-lens depth maps are fused using a semi-global regularization process. They further incorporate a semi-global coarse regularization for insufficiently textured scenes.

In a different approach, Tao *et al.* [12] used a focal stack to estimate depth in a depth-from-defocus approach, by simultaneously using defocus and correspondences. They combine both cues using a Markov random field framework. Going deeper into the focus, Lin *et al.* [6] proposed, most recently, an approach based on the symmetry of the focal stack to estimate depth. They prove that the focal stack is symmetric centered in the in-focus slice, for non-occluded pixels. Occlusions are also studied by Wang *et al.* [13]. They identify the occlusion edges, most useful for object segmentation and, hence, to improve the depth estimation quality. They prove that points in the edge of objects in different depth planes do not meet the standard photometric consistency equation and they derive new expressions for these points.

As for the image rendering, it consists in converting the plenoptic image into a focused image the same way as a conventional camera would see the world. Although the works presented by Ng et al. [9] and Lumsdaine and Georgiev [8] are fast, they present many artifacts and low resolution. Another approach and the one that achieves the best results for multi-focus LF cameras is proposed by Perwass and Wietzke [11]. Having a scene dense depth map it is possible to back trace each pixel onto the image plane. This method allows the render of a high resolution image with few artifacts which can be achieved with a multi-focus plenoptic camera. The major drawback is the high computational power required to process the dense depth map and the image rendering.

In this paper we present a novel fully automatic algorithm to estimate a dense depth map from a single image of a multi-focus plenoptic camera. Our algorithm rely on a robust search for photometric similarity between micro-lenses and by smart mixing images with different levels of blur. The obtained point cloud is

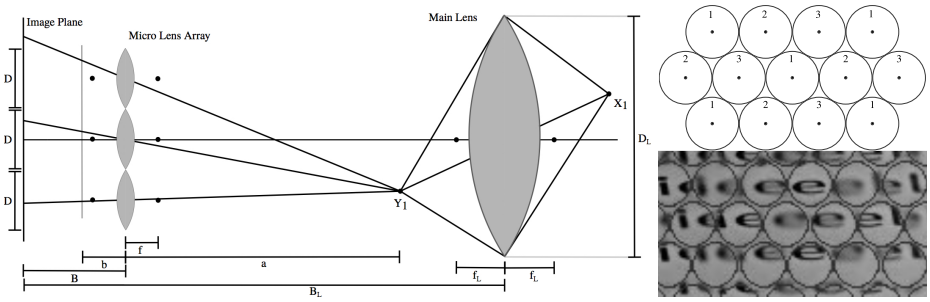


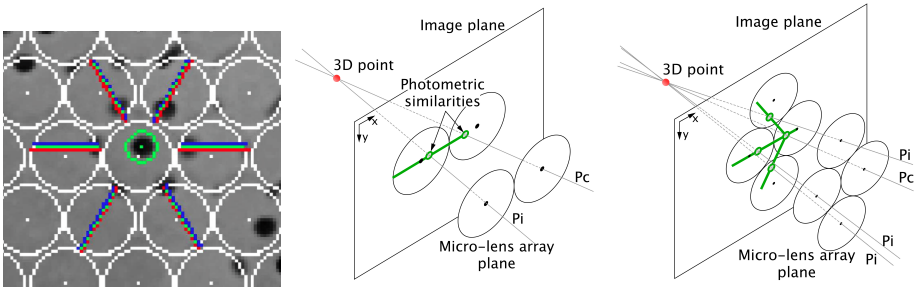
Fig. 1: (a) Plenoptic camera model. (b top) Hexagonal layout of lenses where the lens type is identified by a number. (b bottom) Sample from a Raytrix dataset with different blurs in different lens types.

then filtered to improve the final depth map. We achieve very good results when compared to the state of the art in a considerable less computational time.

2 Multi-focus Plenoptic Cameras

A multi-focus plenoptic camera has a micro-lens array placed in front of the image sensor where each micro-lens have a different focal length from its neighbor lenses. In this paper we are interested in the model presented in [11] and which is represented in figure 1a. For this type of cameras a real world object X_1 (figure 1a) is projected through the camera's main lens onto a virtual image Y_1. This virtual image is then projected through the micro-lens array into the image plane, capturing multiple views of the object. There are lenses with three different focal length, allowing to obtain a larger depth of field. Lenses with different focal lengths will present different blurs for the same depth and will be in focus for different depth ranges. The most common lens type arrangement is hexagonal, as illustrated by the top image of figure 1b. The bottom image of figure 1b shows a sample of a scene at a constant depth where it is possible to identify different lens types through blur.

To clarify the different types of depth maps, notice that we define three different concepts: (1) sparse depth map - it is the raw depth map obtained by projecting the 3D virtual points to the image plane and attributing a depth value for each projected pixel, (2) coarse depth map - it is the depth map obtained by attributing a single depth value for each micro-lens - it is a dense map, since all pixels have a depth value, but it is not dense in a conventional camera point of view (it is not a scene's depth map) and (3) dense depth map - it is the scene's depth map obtained by synthesizing the image and attributing a depth value for each pixel, as if the camera were a conventional pinhole one.



(a) **(b)** **(c)**

Fig. 2: (a) Model tested by the RANSAC-like algorithm. The green circle is the salient point and the red, green and blue lines are the epipolar band where we search for correspondences. The green epipolar line is the main test line while the red and blue lines represent the ± 1 pixel tolerance. (b) and (c) 3D representation of the epipolar geometry between two and four micro-lenses respectively. The green line is the epipolar line, the green circles are best photometric similarities and the red dot it the estimated 3D point for the detected similarities.

2.1 Feature Detection and Depth Estimation

Our algorithm to estimate a sparse depth map is based on photometric similarities between pairs of micro-lens images. Fleischmann and Koch [3] use a similar approach, based on photometric similarity. We use SIFT to search for salient points in the image (as a whole). This method allows us to obtain the most significant points in the image only by adjusting threshold parameters. Regard that we use SIFT features for simplicity and since they have good discriminatory capabilities, however, any salient points detection can replace the use of SIFT. Having the salient points, neighboring lenses are then searched for photometric correspondences, by relying on stereo epipolar geometry (notice that salient points obtained in non useful areas, for instance areas between micro-lenses images, are discarded). Since we are provided a big number of salient points and their respective correspondences, we apply a RANSAC-method to obtain the best 3D point cloud. Our algorithm then back projects the pairs of correspondences. Notice that the distance from the micro-lens array and the image plane is provided by the camera manufacturer (calibration data), allowing to obtain a sparse 3D point cloud. We summarize our method as follows:

– **Step 1 - Selection of the epipolar lines**. For each salient point within a reference micro-lens image I_a, a subset of epipolar lines are considered based on a group of target micro-lens images $I_{a_1}, ..., I_{a_n}$ (neighbor micro-lenses). The n epipolar lines for a point x in the reference image are given by $L_i = \{x + t\upsilon : t \in \mathbb{R}\}$ with $\upsilon = (c_{a_i} - c_a)/2r$ [3], where c_{a_i} and c_a are the coordinates of the center of the target micro-lens a_i with $i \in \{1, ..., n\}$ and the reference micro-lens a with r radius respectively. See figure 2a.

– **Step 2 - Find a correspondence**. Photometric similarities are searched within the target micro-lenses along the n epipolar lines for possible disparities $dj \in [0, d_{max}], d_{max} < 2r$. So, it is calculated the sum of absolute differences (SAD) (of equation (1)) [3] between local neighborhoods $\Omega(x)$, in the reference image, and $\Omega(x - d_j\upsilon)$, in the target image.

$$SAD(x, d_j; a, a_i) = \frac{1}{A(x, \upsilon, d_j)} \sum_{u \in \Omega(x)} |I_a(u) - I_{a_i}(u - d_j\upsilon)| \mathbb{1}(u - d_j\upsilon)\,. \quad (1)$$

with

$$A(x, \upsilon, d_j) = \sum_{u \in \Omega(x)} \mathbb{1}(u - d_j\upsilon)\,. \qquad \mathbb{1}(x) = \begin{cases} 1 & \text{if } ||x|| < r \\ 0 & \text{else} \end{cases}\,.$$

By minimizing the SAD through equation (2) it is obtained the pixel coordinates for the best photometric similarity within each epipolar line of the neighbor micro-lenses.

$$X(a, a_i) = \underset{x}{\operatorname{argmin}}\, SAD(x, d_j; a, a_i), \qquad s.t.\ X(a, a_i) \in I_{a_i} \quad (2)$$

– **Step 3 - Estimation of the 3D virtual points**. A subset of lines are defined, representing one pixel tolerance for the epipolar line (figure 2a), and are grouped two by two. For each pair it is computed the 3D point that minimizes the distance (in 3D) between lines P_i and P_c (figure 2b). The final 3D point has the median of their coordinates
– **Step 4 - Testing the model**. Having an hypothetical 3D point obtained in the previous step, we now need to test the hypothesis for this virtual point. The chosen error measurement is the average distance of the virtual candidate points to the correspondent lines obtained in the previous step.
– **Step 5 - Assessment of the model**. A threshold on the measure defined on the previous step is defined so that only the best estimations are selected. This allows to assume which lines are suited to add to the model (inliers). If there is more than one outlier, the hypothetic model is discarded and we go back to step 1.
– **Step 6 - Re-estimations of the 3D virtual point**. This step is similar to step 3. We re-estimate all 3D virtual points using only the inliers. These lines are again grouped two by two and the 3D point for every combination is the point that minimizes the distance between them. The final 3D point is the median coordinates of all points generated by every line combination.
– **Step 7 - Error metrics**. In this step we evaluate the model in terms of error. It is a mean error from the inliers's distances obtained in step 3, as well as the number of neighbor micro-lenses where a correspondence is found. This will be further discussed on section 2.2.
– **Step 8 - Repeat steps 1-7 for every salient point**.

The output of the previous algorithm is a 3D point cloud of virtual points as projected by the main lens of the camera to their virtual image. At a final stage,

a coarse regularization method will reproject the 3D points of the cloud to the micro-lens images and, thus, attribute an average depth for every micro-lens.

As for the lens pattern used in step 1 (where neighbor lenses are searched for replications of a given salient point) we use different combinations of lenses. Knowing that for a multi-focus plenoptic camera there are lenses with different types, we define lens groups based on the lens type and the distance to the central lens. Figure 3 shows these configurations. We do a smart mixture of lens groups that, even mixing different blurs due to the different focal lengths, is able to optimize the depth estimated throughout the scene's depth ranges. Notice that the depth accuracy depends on the stereo baseline, which is smaller for farther scene depths. Our smart adaptive mixture of micro-lens is able to adjust baseline and range. The neighborhood is limited to R_5 because there is no major correspondences beyond this distance to the center lens (about $3.5D$). Table 1 summarizes all lens patterns studied in our work, where D is the lens diameter.

2.2 Depth improvement

Assume z as the virtual depth of a generic point of the captured scene. As stated by Perwass and Wietzke [11], the maximum radius (R_{max}) that determines the number of micro-lenses that replicate a certain feature is given by equation (3), where B is the distance between the micro-lens plane and the image plane and D is the micro-lens diameter (in pixels). Consequently, the closer a point is to the camera (higher virtual depth), the more lenses will replicate it. Figure 4 is an example for both close and far features on a raw image. When using the R_0 lens pattern (see figure 3) the algorithm searches adjacent lenses with different types for feature matching, being adequate for farther depth ranges. On the other hand, the R_1 configuration is always adequate since it searches lenses of the same type (same focal length). Notice that, as for the R_5 pattern, although the number of correspondences obtained is much lower, their baseline is much higher

Fig. 3: Illustration of the lens neighborhood, with every group labeled from R0 to R5, and lens type from 0 to 2.

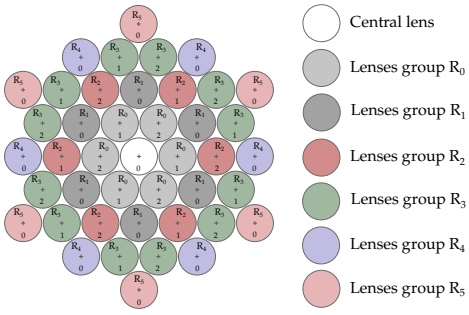


Table 1: Table summarizing the lens pattern parameters (assumes that the central lens type is 0).

Lense Patterns	# Of Lenses	Lenses Types	Distance to central micro-lens
R_0	6	1, 2	D
R_1	6	0	$\sqrt{3} \times D$
R_2	6	1, 2	$2 \times D$
R_3	12	1, 2	$\sqrt{7} \times D$
R_4	6	0	$3 \times D$
R_5	6	0	$2\sqrt{3} \times D$

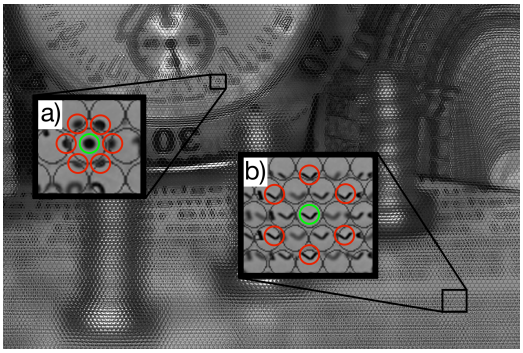


Fig. 4: Salient point replication on neighbor lenses: a) feature replication for a low z value sample. b) feature replication for a high z value sample

and, therefore, the back projection of its correspondences is more stable. Our algorithm then presents an adaptive mixture of micro-lens patterns, by using as many information as possible, and selecting the more stable configurations when available.

$$R_{max} = \frac{|z| \times D}{2 \times B} \tag{3}$$

Lens selection On seeking a more precise reconstruction of the depth map we propose the aggregation of both R_0, R_1, R_4 and R_5 depth data points by quartile sectioning and weight attribution. Our work focuses on the usage of R_0, R_1, R_4 and R_5 lens configuration since they produce the majority and most consistent results of all configurations. Since correspondences in R_4 and R_5 are not always available (only for very close points relative to the camera position) and their blur is similar to the blur of correspondences in R_1, we mention the union of both configurations as $R_1 + R_4 + R_5$. For the fusion of R_0 and $R_1 + R_4 + R_5$ depth maps we consider a linear combination of their estimated depths, given by equation (4), where α is the weight parameter, varying between 0 and 1 based on the depth of the corresponding depth point. We divided the depth map range (from the sparse point cloud) into quartiles so that the first quartile represents the closer depth data points and the fourth quartile the farther depth data points relative to the camera position. For the first quartile the depth data points are extracted from the $R_1 + R_4 + R_5$ depth map, being $z = z_{R_1+R_4+R_5}$. The same applies for the fourth quartile, being $z = zR_0$. For the second and third quartile we use a linear weight of both depth maps.

$$z = \begin{cases} z_{R_1+R_4+R_5}, & \text{if } \hat{z} \in Q1 \\ (1-\alpha)z_{R_0} + \alpha z_{R_1+R_4+R_5}, & \text{if } \hat{z} \in Q2 \cup Q3 \\ z_{R_0}, & \text{if } \hat{z} \in Q4 \end{cases} \tag{4}$$

where $\hat{z} = \frac{z_{R_0}+z_{R_1+R_4+R_5}}{2}$.

Detection and Correction of Highly Blurred Areas. Some plenoptic images might contain sections where none of the lens types can focus. In these cases it is more favorable to assume z = $z_{R_1+R_4+R_5}$ rather than use highly blurred lenses for the final depth map. The texture in these far depth sections is blurred and it is hard to find salient points since they highly depend on texture detail. Then, the algorithm might detect salient points and correspondences for a minimum solution of the RANSAC-like algorithm (when only two correspondences are detected), which will result in a less accurate depth map. For these cases the estimated depth points are not consistent and assume a noisy representation (overfitting). For a non-minimum solution (when more then two correspondences are used) the depth map is not dense enough for a sufficiently dense reconstruction. To solve this problem we generate a minimum solution depth map and we cross it with the non-minimum (robust) solution depth map of the same plenoptic image. For the lack of space we omit the details.

2.3 Coarse Depth Map

For the reconstruction of the dense depth map we use a coarse depth map, having one depth per micro-lens. This process is related with the reprojection of the sparse map points from the source virtual object onto the image plane through the center of the micro-lenses. First, we have to identify which features of the sparse point set are projected through each micro-lens. Even though we do not have the focal length value for each micro-lens, since the distance between image plane and micro-lens array is known, we project every feature within the cone centered on every micro-lens and with radius R_{lens} (figure 5). This is of key importance since even without calibration of the lenses we are able to reconstruct depth. The lens depth is estimated by averaging the depth values of the point set projected into its R_{lens} radius. Notice that a point can be projected through several micro-lens depending on its virtual depth. For each set of points

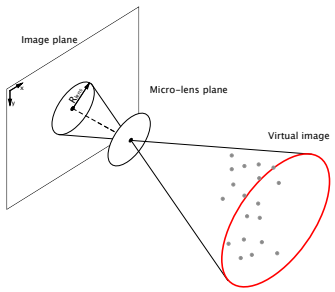


Fig. 5: R_{lens} projection cone for one micro-lens and features that fall inside it.

Table 2: Computational time (in seconds) and mean absolute disparity error (in pixels) for both our and Fleischmann and Koch [3] algorithm and root mean square error for our dense depth map.

	Computation time					MAE		RMSE
Datastes	Our sparse est.	Our coarse est.	Our dense est.	Our total	F&K total	Our coarse est.	F&K est.	Our dense est.
Bunny	368s	710s	77s	**1155s**	3874s	0.497	**0.195**	3.4%
Bolt	450s	855s	80s	**1385s**	4473s	0.271	**0.174**	2.9%
4plane	556s	1109s	85s	**1750s**	4300s	0.230	**0.178**	2.5%

projected into each micro-lens a fine filter is applied. This filter allows a more robust estimation for the depth of each micro-lens, being this depth the averaging of every point's color intensity that follows equation (5) for a local median $\hat{p}$ and standard deviation σ_p of $P(n)$ (local point set with n points) where Ω_p is the point set domain. We then obtain a single depth per micro-lens given by $Z(a_i)$.

$$P_{filtered} = \{P(n) : P(n) \in [\hat{p} - \sigma_p, \hat{p} + \sigma_p], n \in \Omega_p\}. \tag{5}$$

To densely fill every micro-lens without depth we propagate its neighbor lens's depth value. The propagated depth is an averaging of the neighbor lenses depth (assuming a robust region growing with three or more neighbor lenses).

2.4 Dense Depth Map

As for the rendering of the dense depth map we use, as basis information, the coarse depth map. This algorithm is based on the synthesization method of Perwass and Wietzke [11] with a modification at the micro-lens selection for the depth estimation. Instead of selecting the micro-lenses inside the R_{max} radius based on their effective resolution ratio (since we avoid to use the focal length of micro lenses), we use all the lenses inside R_{max}. The final depth (or intensity) value is the weighted mean of the depth (or intensity) of the selected micro-lenses, where the weight accounts for the vignetting effect on lenses (notice that, as stated by [5,10], the micro-lenses have a considerable vignetting effect).

3 Results

We compare our results to the method of Fleischmann and Koch [3]. We measure the computational time and the mean absolute error (MAE) for both methods, as shown on table 2. These measurements where performed on synthetic datasets and on real world datasets provided by Raytrix. Our algorithm achieves comparable results to those of [3] with less computation time for the micro-lens coarse map, and additionally estimating a dense depth map. Since Fleischmann and Koch do not produce a dense depth map, we present the root mean square error (RMSE) between our dense depth map and the depth ground truth of the synthetic datasets (table 2). The RMSE for the synthetic datasets is considerably

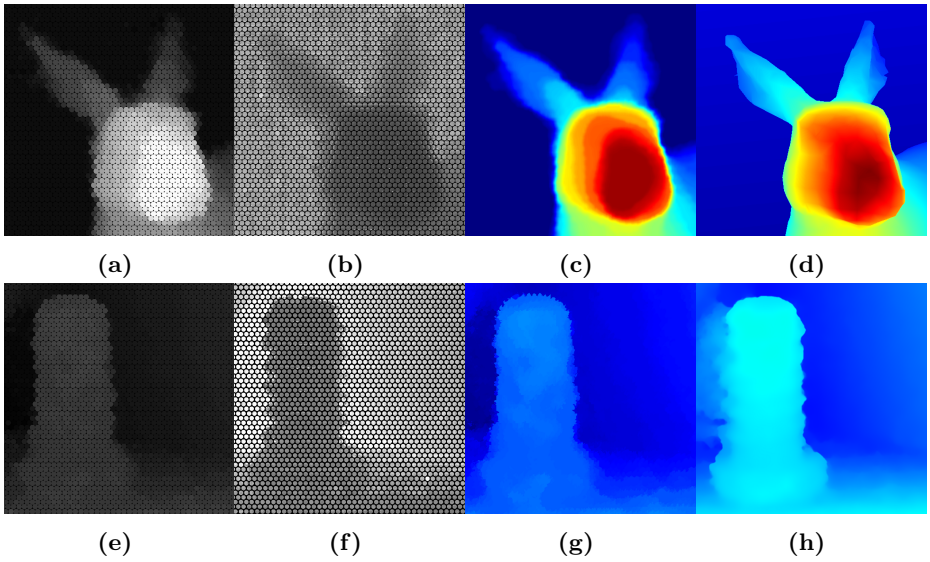


Fig. 6: Excerpts of ours, ground truth and Raytrix's results. (a-e) our coarse depth estimation, (b-f) Fleischmann and Koch's disparity estimation, (c-g) our dense depth estimation, (d) depth ground truth, (h) Raytrix's results.

low but, since Raytrix do not provide the depth ground truth, we present a visual comparison with Raytrix's results, shown in figure 6. Since these images are not immune to aliasing effects due to resize, we present excerpts of the scenes. Detailed full resolution images can be found on the supplementary material. The color pallet we used on our dense depth map is different from Raytrix's. Since we don't know the color pallet used by Raytrix, we applied the OpenCV "colormap_jet" pallet to the scene's depth. Results on several additional datasets are also present in supplementary material.

4 Conclusion

In this paper we propose a light weight algorithm to estimate the depth of a plenoptic image based on detected features for a multi-focus plenoptic camera. Our method uses a coarse map with one depth value per micro-lens to estimate the dense depth map of the captured scene. We test our method on synthetic and real world datasets, comparing them to the method of Fleischmann and Koch [3]. This is an accuracy vs. computation time comparison where our algorithm achieves comparable results in substantial less time. Although our algorithm presents higher error measurements but relatively close to [3], the achieved error values are lower than half of a pixel size. The computation time of our algorithm can still be improved with GPU parallel processing.

References

1. Bishop, T.E., Zanetti, S., Favaro, P.: Light field superresolution. ICCP, IEEE International Conference on Computational Photography pp. 1–9 (2009)
2. Dansearau, D., Bruton, L.: Gradient-based depth estimation from 4d light field. International Symposium on Circuits and Systems 3, III – 549–52 (2004)
3. Fleischmann, O., Koch, R.: Lens-based depth estimation for multi-focus plenoptic cameras. 36th German Conference on Pattern Recognition 8753, 410–420 (October 2014)
4. Ives, H.E.: Optical properties of lippman lenticulated sheet. Journal of the Optical Society of America 21, 171 (1930)
5. Liang, C.K., Ramamoorthi, R.: A light transport framework for lenslet light field cameras. ACM Transactions on Graphics (TOG) 34(2), 16 (2015)
6. Lin, H., Chen, C., Bing Kang, S., Yu, J.: Depth recovery from light field using focal stack symmetry. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3451–3459 (2015)
7. Lippmann, G.: Épreuves réversibles. photographies intégrals. Comptes-Rendus Academie des Sciences 146, 446–451 (1908)
8. Lumsdaine, A., Georgiev, T.: Full resolution lightfield rendering. Indiana University and Adobe Systems, Tech. Rep (2008)
9. Ng, R.: Digital light field photography. Ph.D. thesis, stanford university (2006)
10. Ng, R., Levoy, M., Bredif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Computer Science Technical Report CSTR 2(11) (2005)
11. Perwass, C., Wietzke, L.: Single lens 3d-camera with extended depht-of-field. SPIE Human Vision and Electronic Imaging (2012)
12. Tao, M., Hadap, S., Malik, J., Ramamoorthi, R.: Depth from combining defocus and correspondence using light-field cameras. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 673–680 (2013)
13. 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. pp. 3487–3495 (2015)
14. Wanner, S., Goldlueck, B.: Globally consistent depth labeling of 4d light fields. Computer Vision and Pattern Recognition, 2012 IEEE Conference on pp. 41–48 (2012)