Accurate and fast micro lenses depth maps from a 3D point cloud in light field cameras

Rodrigo Ferreira
Institute of Systems and Robotics
University of Coimbra, Portugal
rferreira@isr.uc.pt

Nuno Gonçalves
Institute of Systems and Robotics
University of Coimbra, Portugal
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 method for depth estimation from a single plenoptic image running a RANSAC-like algorithm for feature matching. The novelty about our method is the global method to back project correspondences found using photometric similarity to obtain a 3D virtual point cloud and different methods to build a depth map from the 3D point cloud generated. We use lenses with different focal-lengths in a multiple depth map refining phase, generating a dense depth map. Tests with simulations and real images are presented and compared with the state of the art, showing comparable accuracy for substantial less computational time.**

I. Introduction

Light field 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 (but slow) 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 [1] 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 [2] 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.

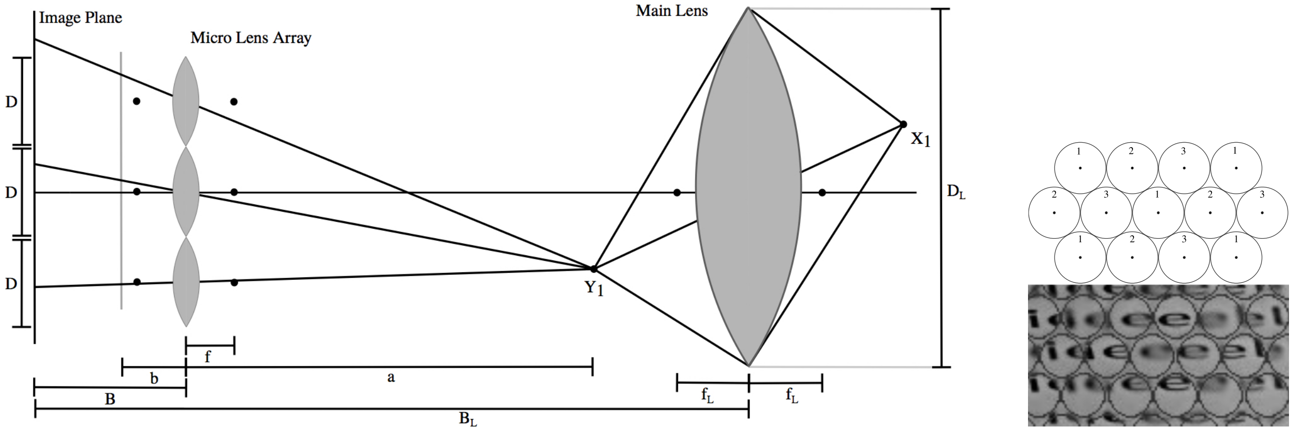
In 2004 Dansearau and Bruton [3] proposed a method for depth estimation using 2D gradient operations. They were able to define the light field direction and thus the depth of the corresponding elements within the light field. The areas where the depth could not be estimated were filled by applying region growing. Since plenoptic cameras are not immune to spatial aliasing, which can result on depth estimation errors, in 2009 Bishop and Favaro [4] applied a different approach to compensate the present aliasing, allowing to recover the depth map from multiple views provided by the 4D light field.

Wanner and Goldluecke [5] 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 into 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 [6] 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.* [7] 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.* [8] 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.* [9]. 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.

Recently, Jeon et al. [10] presentedd a method for depth map estimation based on finding correspondences in sub-aperture images to build a cost volume for optimization. For weak textured regions they use a propagation method to regularize the depth map.

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. [11] and Lumsdaine and Georgiev [12] 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 [13]. Having a scene dense depth map it is possible to back trace each pixel into the image plane. This method allows the



(a)

(b)

Fig. 1: (a) Plenoptic camera setup. (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.

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 micro-lens depth maps 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 then filtered to improve the final depth map. We are, in this paper, particularly interested in comparing different approaches to create a fast, yet accurate, micro-lens depth map from the 3D point cloud obtained by back tracing correspondences. We achieve comparable results, in terms of accuracy, when compared to the state of the art in a considerable less computational time.

II. 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 [13] 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 into 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 depth values 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 as if the scene was taken by a conventional pinhole camera) and (3) dense depth map - it is the depth map obtained by replacing the photometric information with depth information in a conventional camera point of view. This dense depth map is closely related to the all-in-focus renderization of a light field camera. In this paper we are particularly interested in comparing different approaches to obtain the coarse depth map from the point cloud, as stated in the introduction section.

III. Depth Estimation

A. Feature Detection

Our algorithm to estimate a sparse depth map is based on photometric similarities between pairs of micro-lens images. Fleischmann and Koch [6] use a similar approach, based on photometric similarity. We use SIFT to search for salient points in the image. 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. 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:

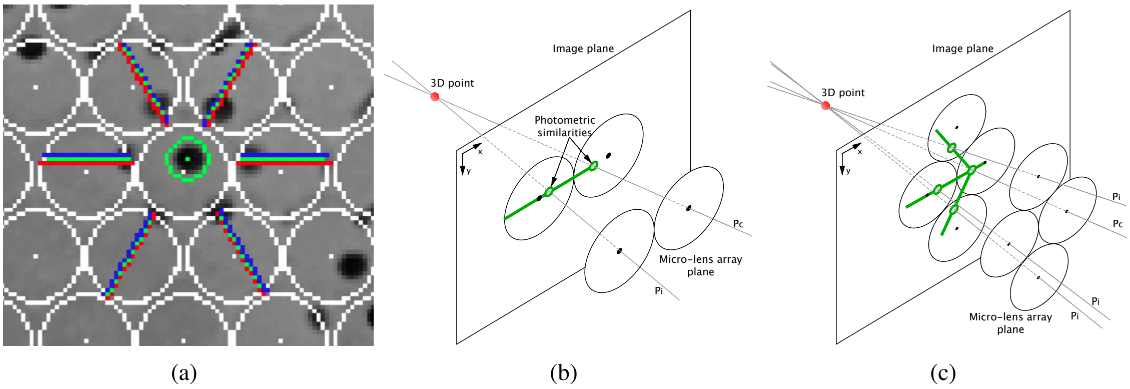


Fig. 2: (a) Model tested by the RANSAC-like algorithm. The green circle is the salient point and the red, green and blue lines are the epipolar band where we search for correspondences. The green epipolar line is the main test line while the red and blue lines represent the ± 1 pixel tolerance. (b) and (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.

- **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 $\upsilon = (c_{a_i} - c_a)/2r$ [6], 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.
- **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)) [6] between a local neighborhood $\Omega(x)$ in the reference image and a local neighborhood $\Omega(x - d_j\upsilon)$ in the target image.

$$SAD(x, d_j; a, a_i) = \frac{1}{A(x, \upsilon, d_j)} \sum_{u \in \Omega(x)} H(u, d_j; a, a_i) \,. \tag{1}$$

with

$$H(u, d_j; a, a_i) = |I_a(u) - I_{a_i}(u - d_j\upsilon)| \mathbb{1}(u - d_j\upsilon) \,.$$

$$A(x, \upsilon, d_j) = \sum_{u \in \Omega(x)} \mathbb{1}(u - d_j\upsilon) \,.$$

$$\mathbb{1}(x) = \begin{cases} 1 & \text{if } ||x|| < r \\ 0 & \text{else} \end{cases} \,.$$

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

$$X(a, a_i) = \operatorname*{argmin}_{x} SAD(x, d_j; a, a_i) \tag{2}$$

- **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 between 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 distance of the virtual candidate point to all the correspondence lines obtained in the previous step.
- **Step 5 - Assessment of the model**. A threshold is defined so we can distinguish the good from the bad estimations. This allows to assume which lines are suited to add to the model (labeled as inliers). If there is more than one outlier, the model is discarded and we go back to the first step. If not, we advance to step 6.
- **Step 6 - Re-estimations of the 3D virtual point**. This step is similar to step 3. We re-estimate the 3D virtual point 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. It is also possible to evaluate the model by the number of neighbor micro-lenses where a correspondence is found.
- **Step 8 - Repeat steps 1-7 for every salient point**.

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 focal lengths, we define lens groups based on the lens type

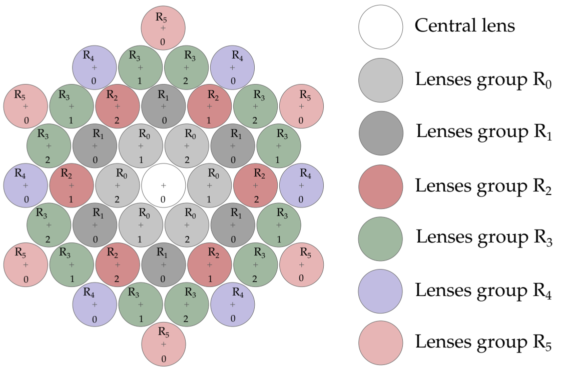


Fig. 3: Illustration of the lens neighborhood, with every group labeled from R0 to R5, and lens type from 0 to 2. The lower value in the micro-lens illustration is the lens type.

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 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.

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, build a micro-lens depth map based on the depths projected. This topic is the main topic of this paper and is further discussed in the next subsection.

B. Coarse Depth Map

The coarse depth map is a 2D depth map where each micro-lens has a depth representation of its projected scene. We use three methods for the reconstruction of the coarse depth map: a depth map with one depth per micro-lens, another with two depths per micro-lens and finally a depth map with a surface fitting for each micro-lens. Regardless of the method, we have to identify which features of the point set estimation are projected through each micro-lens. Even though we do not have the focal length value for each micro-lens, we project every feature within the cone centered on every micro-lens and with radius R_{lens} (this is of key importance since even without calibration of the lenses we are able to reconstruct depth). The R_{lens} projection is illustrated in figure 4.

1) Single Depth Per Micro-lens: The reconstruction of this coarse depth map is related with the reprojection of the sparse map points from the source virtual object into 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. For each set of points 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 (3) for a local median $\hat{p}$ and standard deviation σ_p of $P(n)$ (local point set depth with n points) where Ω_p is the point set depth domain. We then obtain a single depth per micro-lens.

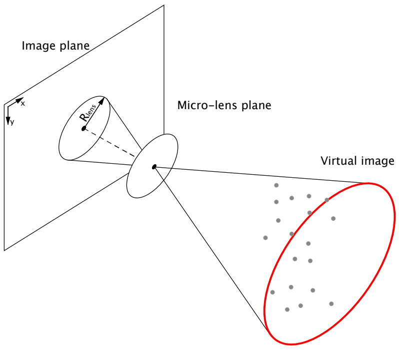


Fig. 4: Generic illustration of R_{lens} radius projection cone for one micro-lens and features that fall inside it.

$$P_{filtered} = \{P(n) : P(n) \in [\hat{p} - \sigma_p, \hat{p} + \sigma_p], n \in \Omega_p\}\,. \quad (3)$$

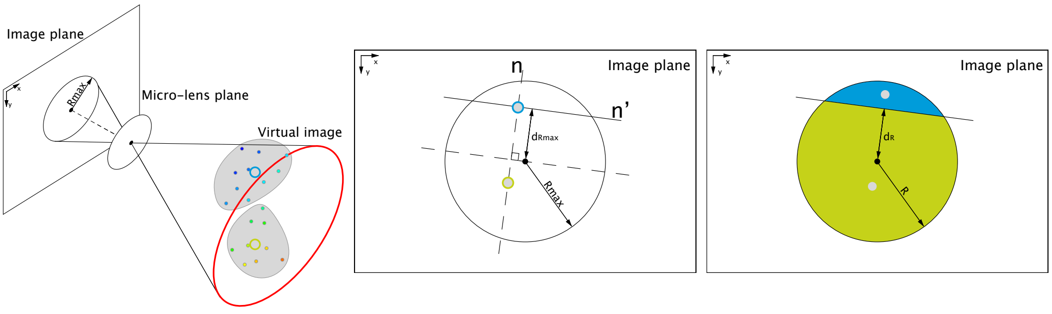
To densely fill every micro-lens without depth information we use a propagation algorithm to its neighbor lens's depth value. The propagated depth is an averaging of the neighbor lenses depth, since it is assumed a robust region growing when there are at least three neighbor lenses with depth information.

2) Two Depths Per Micro-lens: One of our approaches to improve the depth estimation is the sectioning of the micro-lens into two depths (we aim at investigating if the simple and fast approach of using only two depths can per si enhance the accuracy). For this we use the clusterization algorithm k-means [14], which is a self-learning (loop) algorithm based on vector quantization. This method classifies a $N-$dimension point set through k number of clusters. The main objective is to find k centroids, each representing the center of a group of points.

This method allows us to separate the point set into groups. Since it is sensitive to the initial randomly selected cluster centers, it can be triggered several times to reduce the error effect of the random initial conditions (we use the OpenCV implementation, which is optimized to multithreading).

As for the micro-lens sectioning, similarly to the single depth per lens approach, we identify which points fall inside the projection cone with a radius R_{max} for each micro-lens. Having a local point set for each micro-lens, without projecting these points, we group them into 2 clusters and extract their centers. This is illustrated in figure 5a.

Following, the clusters centers are projected into the image plane through the micro-lens center, assigning them a color intensity value of their respective virtual depth. As seen in figure 5b, the 2D line that sections the micro-lens (illustrated as n') intersects the center of the projected cluster that maximizes the distance to the center of the micro-lens. This line is normal to the line that intersects the centers of the projected clusters



(a) (b) (c)

Fig. 5: (a) R_{max} projection cone, features that fall inside it (dense colored points) and cluster's center (border colored circles). (b) Clusters and R_{max} projection on the image plane. Normal line (continuous line) passing through the farthest cluster relative to the micro-lens projected center. (c) Sectioned micro-lens with assigned depths equal to the clusters virtual depth.

(illustrated as n). The n' line's equation is then normalized and scaled to the radius of the micro-lens. The assigned depth for both partitions is equal to the average depth values of the assigned cluster points (figure 5c).

The micro-lens sectioning only occurs if the two clusters present a significant depth difference. Otherwise, we assume a single depth per micro-lens.

3) Micro-lens Surface Fitting: Since one knows the depth of the points reprojected to the micro-lens image, it is a natural choice to fit these points depth to a surface (3D reconstruction for each micro lens). We use a robust least squares approach. This method is simple and its accuracy increases with the size of the point set.

As stated by Ambrosius [15], given a set of n points $x_1 \dots x_n, y_1 \dots y_n$, with corresponding $z_1 \dots z_n$, it is possible to find a polynomial of degree p (arbitrary) that fits the data with a minimum error in the least squares sense.

The generic polynomial is given by equation (4). We can write this equation in matrix notation as shown in equation (5). The left most matrix (X) is called the Vandermonde matrix.

$$z = a_1 + a_2 x + a_3 y + a_4 x^2 y + a_5 x y^2 + a_6 x^2 y^2 + \dots + a_{(2p+2)} x^p y^p \tag{4}$$

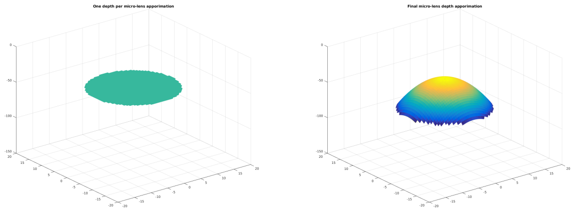
$$Xa = z \tag{5}$$

Knowing each point's x, y and z coordinates and seeking a polynomial of degree $p = 2$ we can easily determine the coefficients a by inverting the Vandermonde matrix V (using the Moore-Penrose pseudo-inverse). By densely re-sampling a micro-lens, it is possible to reconstruct its surface with the second order surface fitting coefficients.

The least squares approximation is a fair approach when the local micro-lens point set is dense, otherwise it might generate inaccurate data for these less dense point sets.

IV. Results

We compare the results of our method to the one of Fleischmann and Koch [6]. We test both methods with synthetic datasets produced by our simulator [16] and real world datasets provided by Raytrix. Table I shows the measures computation time and mean absolute error for all the tested methods. From this table we can see that our methods achieve comparable accurate results with substantially less computation time. From our methods, the two depths per micro-lens method presents the best results and the one depth per micro-lens presents the best computation time. We can see that there is a trade-off between computation time and accuracy for the tested algorithms. Detailed results are presented in supplementary material.



(a) (b)

Fig. 6: Generic example depth estimation of a micro-lens. (a) Single depth per micro-lens estimation for the local micro-lens point set. (b) Least squares approximation surface for the local micro-lens point set.

V. Conclusion

In this paper we propose three new low-cost algorithms to estimate the depth of a plenoptic image based on detected features for a multi-focus plenoptic camera. Our methods generate a coarse depth map with one depth, two depths and a surface fitting per micro-lens. We test our methods on synthetic and real world datasets, comparing them to the method of Fleischmann and Koch [6]. This comparison shows a compromise between computation time and accuracy for the tested methods, where our algorithms achieves comparable accurate results in substantially less computation time. With our method we can estimate depth, even without the calibration data of the micro-lens array, while Fleischmann and Koch only

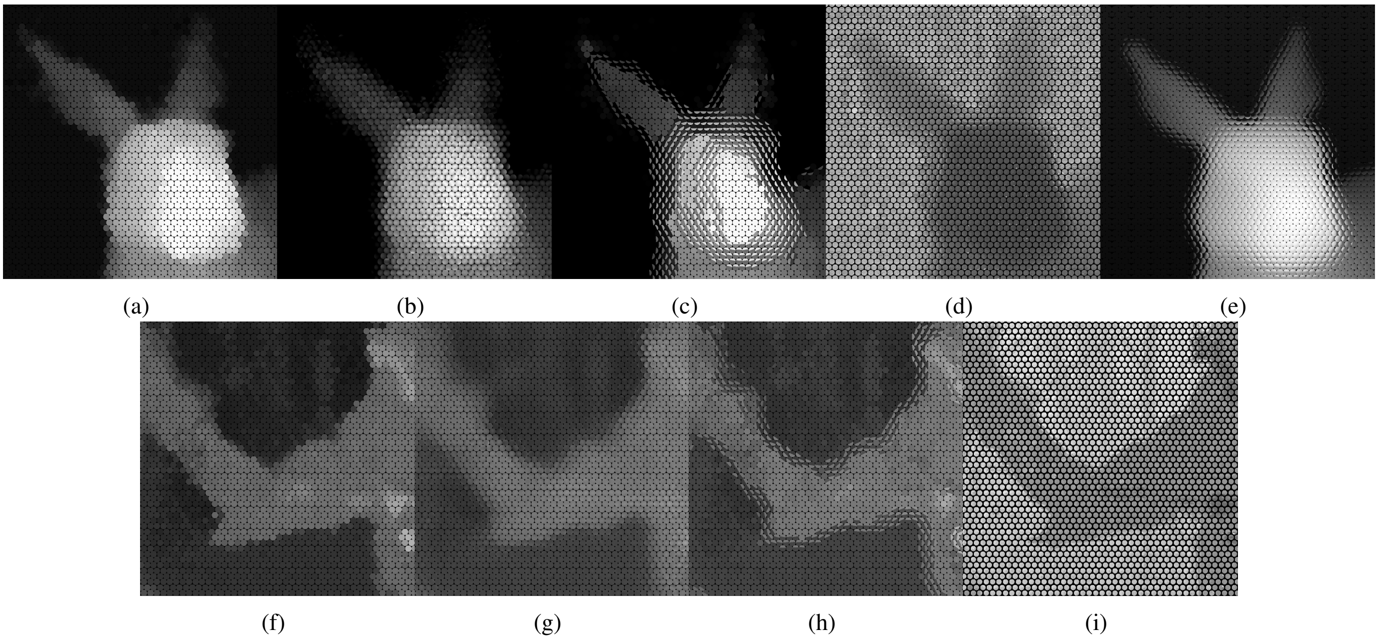


Fig. 7: Results for the synthetic Bunny dataset and Raytrix's Watch dataset (excerpts) with the ground truth. (a/f) our coarse depth estimation with one depth per micro-lens, (b/g) our coarse depth estimation with surface fitting per micro-lens, (c/h) our coarse depth estimation with two depths per micro-lens, (d/i) Fleischmann and Koch's disparity estimation, (e) ground truth.

TABLE I: Computational time (in seconds) and mean absolute error (in pixels) for both our and Fleischmann and Koch [6] algorithms

	Computation time				MAE			
Datasets	Our single depth per micro-lens	Our two depths per micro-lens	Our micro-lens surface fitting	F&K	Our single depth per micro-lens	Our two depths per micro-lens	Our micro-lens surface fitting	F&K
Bunny	**1078s**	2172s	2062s	3874s	0.497	**0.384**	0.388	**0.195**
Bolt	**1305s**	2668s	2119s	4473s	0.271	**0.190**	0.197	**0.174**
4plane	**1665s**	2989s	2314s	4300s	0.230	**0.217**	0.231	**0.178**

estimate disparities. The computation time of our algorithm can still be improved with GPU parallel processing

Acknowledgements

The authors acknowledge the support of the Portuguese Science Foundation through grant UID/EEA/00048/2013.

References

[1] G. Lippmann, "Épreuves réversibles. photographies intégrals," *Comptes-Rendus Academie des Sciences*, vol. 146, pp. 446–451, 1908.

[2] H. E. Ives, "Optical properties of lippman lenticulated sheet," *Journal of the Optical Society of America*, vol. 21, p. 171, 1930.

[3] D. Dansearau and L. Bruton, "Gradient-based depth estimation from 4d light field," *International Symposium on Circuits and Systems*, vol. 3, pp. III – 549–52, 2004.

[4] T. E. Bishop, S. Zanetti, and P. Favaro, "Light field superresolution," *ICCP, IEEE International Conference on Computational Photography*, pp. 1–9, 2009.

[5] S. Wanner and B. Goldlueck, "Globally consistent depth labeling of 4d light fields," *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*, pp. 41–48, 2012.

[6] O. Fleischmann and R. Koch, "Lens-based depth estimation for multi-focus plenoptic cameras," *36th German Conference on Pattern Recognition*, vol. 8753, pp. 410–420, October 2014.

[7] M. 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*, 2013, pp. 673–680.

[8] H. Lin, C. Chen, S. Bing Kang, and J. Yu, "Depth recovery from light field using focal stack symmetry," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3451–3459.

[9] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3487–3495.

[10] H.-G. J. et al., "Accurate depth map estimation from a lenselet light field camera," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] R. Ng, "Digital light field photography," Ph.D. dissertation, stanford university, 2006.

[12] A. Lumsdaine and T. Georgiev, "Full resolution lightfield rendering," *Indiana University and Adobe Systems, Tech. Rep*, 2008.

[13] C. Perwass and L. Wietzke, "Single lens 3d-camera with extended depht-of-field," *SPIE Human Vision and Electronic Imaging*, 2012.

[14] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[15] F. Ambrosius, "Interpolation of 3d surfaces for contact modeling," University of Twente, Tech. Rep., March 2005.

[16] R. Ferreira, J. Cunha, and N. Goncalves, "Multi-focus plenoptic simulator and lens pattern mixing for dense depth map estimation," *Eurographics Short Papers*, 2016.