PlenoptiSign: An optical design tool for plenoptic imaging

Christopher Hahne[1],[*], Amar Aggoun

University of Wolverhampton, School of Mathematics and Computer Science, Wolverhampton, United Kingdom

ARTICLE INFO

ABSTRACT

Plenoptic imaging enables a light-field to be captured by a single monocular objective lens and an array of micro lenses attached to an image sensor. Metric distances of the light-field's depth planes remain unapparent prior to acquisition. Recent research showed that sampled depth locations rely on the parameters of the system's optical components. This paper presents PlenoptiSign, which implements these findings as a Python software package to help assist in an experimental or prototyping stage of a plenoptic system.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Code metadata

Current code version	1.1.2
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2019_14
Legal Code License	GNU GPL-3.0
Code versioning system used	Git
Software code languages, tools, and services used	Python, JavaScript/AJAX
Compilation requirements, operating environments & dependencies	NumPy, Matplotlib, tkinter, ddt, cgi
If available Link to developer documentation/manual	https://github.com/hahnec/plenoptisign/blob/master/README.rst
Support email for questions	info [ät] christopherhahne.de

Software metadata

Current software version	1.1.2
Permanent link to executables of this version	https://github.com/hahnec/plenoptisign/releases
Legal Software License	GNU GPL-3.0
Computing platforms/Operating Systems	OS X, Microsoft Windows, Unix-like, web-based etc.
Installation requirements & dependencies	NumPy, Matplotlib, tkinter
If available, link to user manual — if formally published include a reference to the publication in the reference list	https://github.com/hahnec/plenoptisign/blob/master/README.rst
Support email for questions	info [ät] christopherhahne.de

1. Motivation and significance

Plenoptic cameras gain increasing attention from the scientific community and pave their way into experimental three-dimensional (3-D) medical imaging [1–4]. A limitation of light-field cameras is that the maximum distance between two viewpoint positions, the so-called baseline, is confined to the extent of the entrance pupil [5,6]. With triangulation, small baselines are mapped to depth planes close to the imaging device making them suitable for medical purposes such as in a microscope [1,2] or an otoscope [3].

When capturing depth with a plenoptic system, it is essential to place available light-field depth planes on the targets of interest. Hence, it is a key task in plenoptic data acquisition to choose suitable specifications for the micro lenses, the objective lens and

---

* Corresponding author.
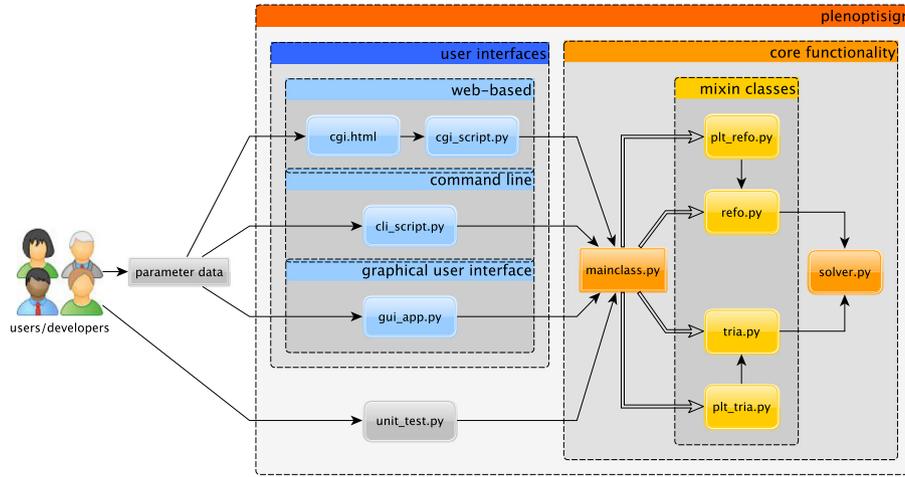[1] email: info [at] christopherhahne.de

Fig. 1. PlenoptiSign's software architecture.

the sensor early to save time and costs at the conceptual design stage of a prototype.

`PlenoptiSign` enables *a priori* depth plane localization in plenoptic cameras for stereo matching from sub-aperture image disparities [5–7] as well as computational refocusing via shift-and-integration [8]. `PlenoptiSign` can be used to pinpoint object distances in light-field images rendered by our complementary open-source software `PlenoptiCam` [9]. The underlying physical model of `PlenoptiSign` was devised and experimentally proven in studies by Hahne et al. [6,10] and applies to the Lytro-type setup [11] at this stage.

Given an experiment involving plenoptic image acquisition, a researcher may want to simulate the influence of the Micro Lens Array (MLA), the objective lens and its focus as well as optical zoom settings to investigate the depth resolution performance. With this software, the user is able to optimize an experimental setup as required. In its current state, the tool can be called from a Graphical User Interface (GUI), a web-server capable of handling the Common Gateway Interface (CGI) or from the Command Line Interface (CLI) where a user is asked for all input parameters.

It is the goal of this paper to raise awareness of the light-field model's capabilities when implemented as a software tool. In Section 2, we sketch the software architecture while turning the focus onto the ray function solver by means of linear algebra to complement previous publications where this has not been explained in much detail. This is followed by usage instructions and exemplary result presentations in Section 3 after which the potential influence on future applications is discussed in Section 4.

2. Software description

2.1. Architecture

Plenoptic camera parameters are passed to `PlenoptiSign` using either the CLI (*cli_script.py*), a `tkinter` GUI (*gui_app.py*) or a CGI web-server (*cgi_script.py*). Based on Python's `ddt` package, a unit test was added to support potential future development. An overview of the code structure is depicted in Fig. 1. An object of the light-field geometry estimator is instantiated from a user interface by calling *mainclass.py*, which inherits mixin classes, namely *refo.py*, *tria.py*, *plt_refo.py* and *plt_tria.py*.

2.2. Functionalities

The core task of this software is to find intersections of light-field ray pairs. Since its implementation has not been entirely covered before [6,10], it is demonstrated hereafter. A concise form of finding an intersection of two ray functions is by solving a System of Linear Equations (SLE) given as

$$\mathbf{Ax} = \mathbf{b} \tag{1}$$

where $\mathbf{A}$ and $\mathbf{b}$ make up ray functions and $\mathbf{x}$ contains unknowns representing locations of ray intersections in the $y-z$ plane. Using two ray functions, we obtain a unique solution to the SLE by the algebraic inverse $\mathbf{A}^{-1}$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{2\times2}$ is invertible. To cover cases of overdetermined SLEs, a more generic solution is provided by the Moore–Penrose pseudo-inverse $\mathbf{A}^+$

$$\mathbf{x} = \mathbf{A}^+\mathbf{b} \tag{3}$$

$$\mathbf{A}^+ = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T} \tag{4}$$

with $^\mathsf{T}$ denoting the matrix transpose. Algebraic computations are implemented by means of the `NumPy` module in the *solver.py* file.

For plenoptic triangulation, matrices $\mathbf{A}$ and $\mathbf{b}$ may be defined according to the notation provided in [6] with Eq. (18) as a ray function, which writes

$$\widehat{f}_{i,j}(z) = q_{i,j} \times z + U_{i,j} , \quad z \in [U, \infty) \tag{5}$$

with $U_{i,j}$ as y-intercepts at the object-side principal plane and $q_{i,j}$ as chosen chief ray slopes in object space. Here, i is an arbitrary micro image pixel index and j represents a micro lens index in one direction. As seen in Fig. 2, we find the separation of two light-field viewpoints, so-called baseline B_G, by two intersecting linear ray functions [6]

$$B_G := \widehat{f}_{i,j}(z) = \widehat{f}_{i+G,j+1}(z) \tag{6}$$

where our reference viewpoint depends on i and is separated by a scalar G of light-field viewpoints. After rearranging the two functions, we write

$$-q_{i,j} \times z + B_G = U_{i,j} \tag{7}$$

$$-q_{i+G,j+1} \times z + B_G = U_{i+G,j+1} \tag{8}$$

which can be solved using the SLE given by

$$\mathbf{A} = \begin{bmatrix} -q_{i,j} & 1 \\ -q_{i+G,j+1} & 1 \end{bmatrix}; \quad \mathbf{x} = \begin{bmatrix} z \\ B_G \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} U_{i,j} \\ U_{i+G,j+1} \end{bmatrix} \tag{9}$$
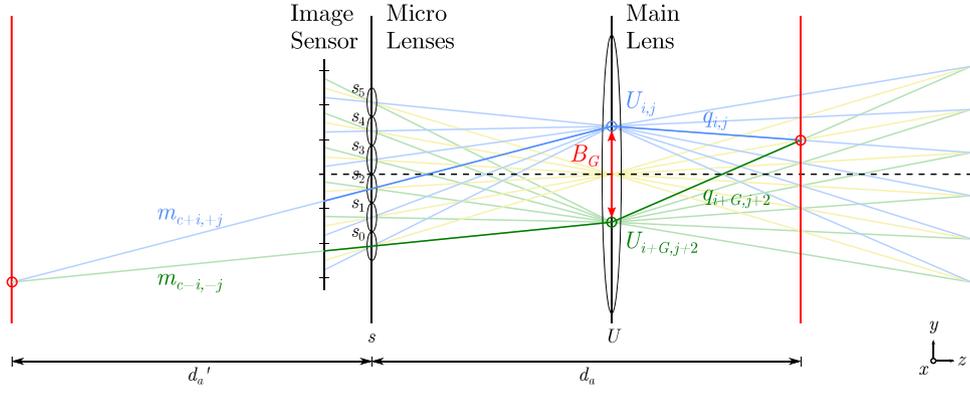
Fig. 2. Plenoptic model with simplified MLA plane s and main lens U along which the baseline B_G ($G = 2$) is found via intersections $U_{i,j}$ from object-side ray slopes $q_{i,j}$. Refocused object distances d_a rely on the image distance extension $d_a{'}$ obtained from image-side ray slopes $m_{i,j}$ and micro lens positions s_j.
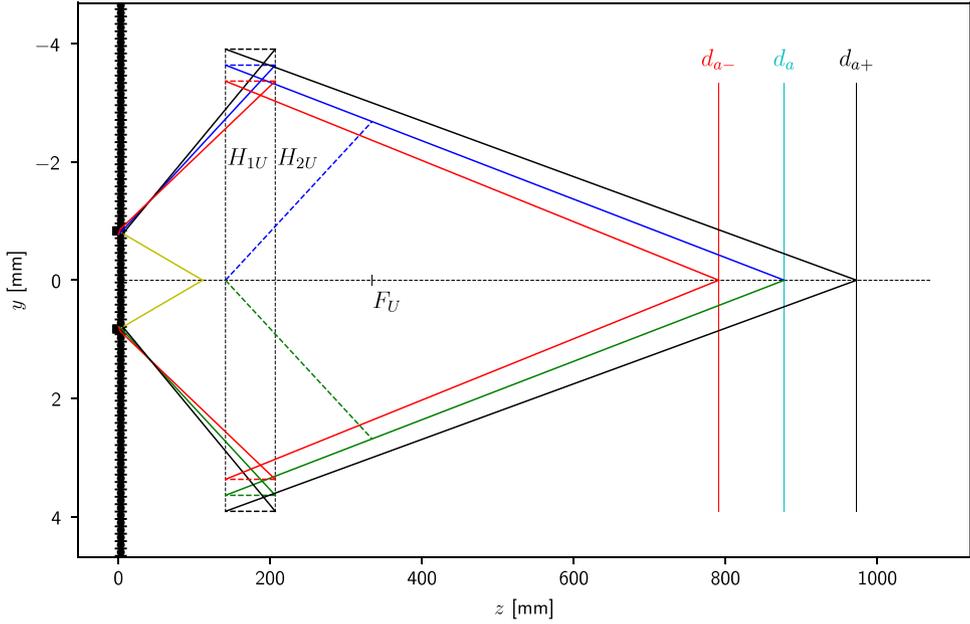

Fig. 3. Refocusing plot for shift parameter $a = 1$ with refocusing distance d_a, depth of field limits $d_{a\pm}$, main lens principal planes H_{1U}, H_{2U} and focal plane F_U.

where z here corresponds to the longitudinal entrance pupil position $\overline{A''H_{1U}}$ [6].

Similarly, we can trace a pair of light-field rays to find an object plane to which a plenoptic image is focused using the shift and integration algorithm [8,10]. In accordance with the model presented in [10], an image-side ray function $f_{c+i,j}(z)$ is given by

$$f_{c+i,j}(z) = m_{c+i,j} \times z + s_j, \quad z \in (-\infty, U) \tag{10}$$

where $m_{c+i,j}$ denotes an image-side ray slope and s_j the respective micro lens position. A ray is chosen by $i = -c$ and $j = a(M-1)/2$ with its counterpart ray having negative indices. Applying this to Eq. (10) and rearranging yields

$$-m_{c+i,+j} \times z = s_{+j} \tag{11}$$

$$-m_{c-i,-j} \times z = s_{-j} \tag{12}$$

which can be represented in matrix form by

$$\mathbf{A} = \begin{bmatrix} -m_{c+i,+j} & 1 \\ -m_{c-i,-j} & 1 \end{bmatrix}; \quad \mathbf{x} = \begin{bmatrix} d_a{'} \\ y \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} s_{+j} \\ s_{-j} \end{bmatrix} \tag{13}$$

where y is the vertical position and $d_a{'}$ is an elongation of the image distance which is mapped to respective refocusing distance d_a using the thin lens equation while taking thick lenses and their principal planes into account [10].

3. Illustrative examples

3.1. Usage

After downloading `PlenoptiSign` from the online repository, installation is made possible with the `setuptools` module which can be run by

   \$ python plenoptisign/setup.py install

in the download directory, provided that Python is available and granted necessary privileges. As indicated in Fig. 1, PlenoptiSign can be accessed from three different interfaces, namely web-based CGI, a GUI and bash command line. Executing PlenoptiSign from the command line is done by

   \$ plenoptisign −p

where −p sets the plot option with rays and depth planes being depicted as seen in Figs. 3 and 4. Starting the `tkinter`-based GUI application is either done by running a bundled executable file or by

   \$ plenoptisign −g

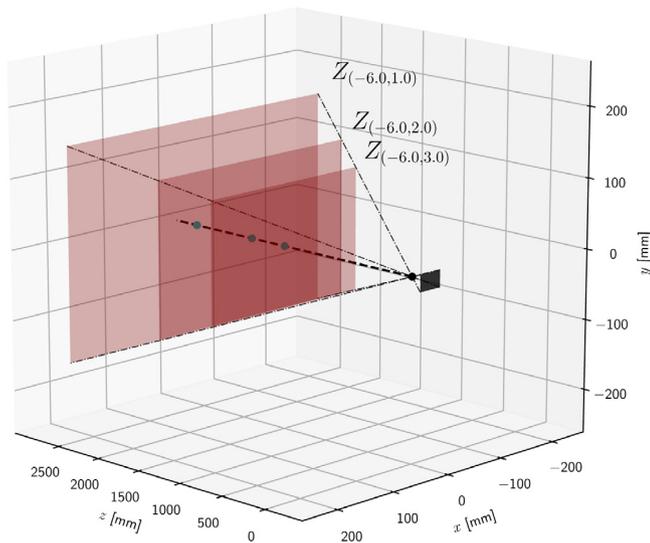For more information on available commands, use the help option −h.

Fig. 4. Triangulation 3-D plot with planes $Z_{(G,\Delta x)}$ and a camera gap $G = -6$.

3.2. Results

Light-field geometry results are provided as text values or graphical plots with two types of views. An exemplary cross-sectional plot is shown in Fig. 3.

In addition, the results can be displayed in 3-D, as depicted in Fig. 4, with triangulation planes $Z_{(G,\Delta x)}$ for disparities Δx. For details on the design trends, scientific notations and deviations due to paraxial approximation, we may refer to our preceding publications [6,10] for further reading.

4. Impact

The presented framework provides answers and generic solutions to a question a researcher raised in a forum [12] and received recommendations from peers, indicating demands and potentials for future projects. Research directions that may benefit from this software include experimental photography [11], cinematography [13] or scientific fields ranging from volumetric fluid particle flow [14] to the many kinds of clinical studies [1–4].

It may be argued that depth plane localization can be done via extrinsic calibration as traditionally employed in the field of stereo vision. Despite its widespread use, this method cannot be performed prior to the camera manufacturing process, but solely after the fact. The same applies to a calibration of light-field metrics via least-squares fitting [15]. Due to the heuristic nature of this approach, zoom lens and optical focus settings as well as temperature fluctuations introduce degrees of freedom to the fitted curve that make calibration sophisticated. After all, it is essential to comprehend and exploit the underlying optical model in the development stage of a plenoptic camera for optimal performance. This part would turn out to be costly with the aforementioned alternatives.

A probably deviant attempt of the presented refocusing distance algorithm has been used in the auspicious Lytro cameras [11]. However, Lytro's method remains undisclosed up to the present day, which once more underlines the need and practical use of our open-source software.

5. Conclusions

Thanks to provided software, the light-field geometry of a plenoptic camera can be accurately predicted with ease of use. This is of crucial interest in the design stage of a prototype where distance range and depth plane density need to be determined and optimized in advance. This tool has been made available for web-servers, a graphical user-interface and the command line tool. It is the first open-source software to do so and may lay the groundwork for future research on plenoptic imaging in the medical field. Future development may lead towards the implementation of an automatic parameter optimization or an extension of a focused plenoptic camera model.

Declaration of competing interest

The authors declare that there is no known conflict of interest.

Acknowledgments

References

[1] Prevedel R, Yoon Y-G, Hoffmann M, Pak N, Wetzstein G, Kato S, Schrödel T, Raskar R, Zimmer M, Boyden ES, et al. Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy. Nature Methods 2014;11(7):727.

[2] Li H, Guo C, Kim-Holzapfel D, Li W, Altshuller Y, Schroeder B, Liu W, Meng Y, French J, Takamaru K-I, Frohman M, Jia S. Fast, volumetric live-cell imaging using high-resolution light-field microscopy. bioRxiv 2018. http://dx.doi.org/10.1101/439315, https://www.biorxiv.org/content/early/2018/10/10/439315.

[3] Bedard N, Shope T, Hoberman A, Ann Haralam M, Shaikh N, Kovačević J, Balram N, Tošić I. Light field otoscope design for 3D in vivo imaging of the middle ear. Biomed. Optics Express 2017;8:260. http://dx.doi.org/10.1364/BOE.8.000260.

[4] W. Palmer D, Coppin T, Rana K, G. Dansereau D, Suheimat M, Maynard M, A. Atchison D, Roberts J, Crawford R, Jaiprakash A. Glare-free retinal imaging using a portable light field fundus camera. Biomed. Optics Express 2018;9:3148. http://dx.doi.org/10.1364/BOE.9.003148.

[5] Adelson EH, Wang JY. Single lens stereo with a plenoptic camera. IEEE Trans Pattern Anal Mach Intell 1992;14(2):99–106.

[6] Hahne C, Aggoun A, Velisavljevic V, Fiebig S, Pesch M. Baseline and triangulation geometry in a standard plenoptic camera. Int J Comput Vis 2018;126(1):21–35. http://dx.doi.org/10.1007/s11263-017-1036-4.

[7] Ng R. Digital light field photography (Ph.D. thesis), Stanford University; 2006.

[8] Isaksen A, McMillan L, Gortler SJ. Dynamically reparameterized light fields. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '00, New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.; 2000, p. 297–306. http://dx.doi.org/10.1145/344779.344929.

[9] Christopher Hahne. PlenoptiCam. 2019, https://github.com/hahnec/plenopticam.

[10] Hahne C, Aggoun A, Velisavljevic V, Fiebig S, Pesch M. Refocusing distance of a standard plenoptic camera. Opt. Express 2016;24(19):21521–40. http://dx.doi.org/10.1364/OE.24.021521, http://www.opticsexpress.org/abstract.cfm?URI=oe-24-19-21521.

[11] Lytro Inc. Lytro ILLUM user manual, 2.1 edn.. 2015.

[12] Sinan Hasirlioglu. Researchgate. 2017, https://www.researchgate.net/post/Does_anyone_know_how_to_estimate_the_depth_in_meters_using_a_Light_Field_Camera_Lytro_Illum.

[13] Aggoun A, Tsekleves E, Swash R, Zarpalas D, Dimou A, Daras P, Nunes P, Soares L. Immersive 3D holoscopic video system. MultiMedia, IEEE 2013;20(1):28–37. http://dx.doi.org/10.1109/MMUL.2012.42.

[14] Deem EA, Zhang Y, Cattafesta LN, Fahringer TW, Thurow BS. On the resolution of plenoptic PIV. Meas Sci Technol 2016;27(8):084003. http://dx.doi.org/10.1088/0957-0233/27/8/084003.

[15] Schima R, Mollenhauer H, Grenzdörffer G, Merbach I, Lausch A, Dietrich P, Bumberger J. Imagine all the plants: evaluation of a light-field camera for on-site crop growth monitoring. Remote Sens 2016;8:823. http://dx.doi.org/10.3390/rs8100823.