Available online at www.sciencedirect.com
ScienceDirect

SoftwareX 5 (2016) 101–106

SoftwareX

www.elsevier.com/locate/softx

Light-Field Imaging Toolkit

Jeffrey Bolan, Elise Hall, Chris Clifford, Brian Thurow*

Advanced Flow Diagnostics Laboratory, Auburn University, AL, United States

Received 6 August 2015; received in revised form 19 May 2016; accepted 25 May 2016

Abstract

The Light-Field Imaging Toolkit (LFIT) is a collection of MATLAB functions designed to facilitate the rapid processing of raw light field images captured by a plenoptic camera. An included graphical user interface streamlines the necessary post-processing steps associated with plenoptic images. The generation of perspective shifted views and computationally refocused images is supported, in both single image and animated formats. LFIT performs necessary calibration, interpolation, and structuring steps to enable future applications of this technology.

Keywords: Plenoptic; Light field; MATLAB

Code metadata

Current code version	*V2.31*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-15-00046
Legal Code License	*GNU General Public License, version 3 (GPL-3.0)*
Code versioning system used	*git*
Software code languages, tools, and services used	*MATLAB*
Compilation requirements, operating environments & dependencies	*Requires MATLAB 2009a or higher on Microsoft Windows or Linux*
If available Link to developer documentation/manual	*See included documentation PDF at* https://github.com/AFDL/LFIT
Support email for questions	thurow@auburn.edu

1. Motivation and significance

Recently, the technique of light-field imaging with a plenoptic camera has emerged [1–4], which is capable of capturing 3D information about a scene in a single snapshot. This technique uses what is known as a plenoptic camera to capture the four-dimensional (4D) light field—the spatial and angular information associated with the distribution of light rays in a volume [5]. A plenoptic camera is identical in most respects to a conventional camera; the key difference lies in the addition of a microlens array. This planar array of microlenses is inserted inside the camera a short distance in front of the image sensor. Light rays passing through the main lens of the camera are focused onto the microlens array. Depending on the incident angle of a given light ray on the main lens, the ray will be focused by the microlens it strikes onto a different pixel on the image sensor. In this way, both the angular and spatial information in a scene are sampled. This combined spatial and angular data can be computationally processed to generate new, unique images of the volume from a single raw plenoptic image. This allows for 3D information to be collected using an otherwise typical camera. The potential applications of this tool are myriad, ranging from microscopy [6] to 3D particle image velocimetry [7] to range finding [8,9].

While most post-processing is largely optional in a conventional camera, raw plenoptic images require significant post-processing to be of any use. A plenoptic camera uses a standard 2D image sensor but encodes the aforementioned spatial and angular data into thousands of microimages formed by the microlens array. A given microimage corresponds to a single spatial sample of the light field, where each pixel within the given

* Corresponding author.
E-mail address: thurow@auburn.edu (B. Thurow).

http://dx.doi.org/10.1016/j.softx.2016.05.004

microimage corresponds to a different angular sample. To extract this information, spatial and angular coordinates must be associated with each pixel in the image. In essence, although a plenoptic camera may be used identically to a traditional camera, special software is required to extract 3D data from the 2D camera sensor. This need for specialized software, which prior to this work was not openly available, is a significant barrier for researchers wishing to utilize a plenoptic camera in their work.

The Light-Field Imaging Toolkit (LFIT) was written to streamline and standardize the processing of raw plenoptic images. Initially, LFIT was created in 2014 for use in the Auburn University Advanced Flow Diagnostics Laboratory (AFDL) by combining several independently developed codes internal to the lab. Since then, LFIT has been enhanced and extended across a number of versions. Today, LFIT is composed of dozens of functions and several thousand lines of code, with a fully functioning graphic user interface (GUI). This suite of functions is now provided to assist other groups working with plenoptic data. This paper provides a high level overview of LFIT and its functionality.

2. Software description

LFIT is a suite of MATLAB functions, each of which facilitates some aspect of the processing of a raw plenoptic image. A single main demonstration script is provided with the package, which invokes the graphical user interfaces (GUI) and calls the various included functions as appropriate. Most users will be most comfortable with the GUI mode, although a scripted batch mode is also included for automated processing. It should be noted that the individual functions can be used independently of the main demonstration script. That is, all the functionality of LFIT can be accessed via the command line or user-written MATLAB scripts that call LFIT functions. For specialized applications, this may be the preferred route.

Broadly speaking, the overall task of processing a raw plenoptic image can be categorized into these steps: Import, Microlens Calibration, Radiance Array Formation, and Computational Imaging. These steps are described in more detail below. Additional practical details are given in the documentation included with LFIT. An exhaustive description of the theory behind light field imaging with a plenoptic camera is beyond the scope of this paper; for an extended discussion of the theory from a practical perspective with regard to LFIT, refer to [10]. Also note that the nomenclature used in LFIT is largely adopted from Ng; see [4,11] for a full description of light field imaging.

2.1. Import

LFIT first requires that several basic parameters be defined before processing any images. These parameters are defined in the first GUI window, shown below in Fig. 1. Specifically, a directory containing calibration image(s) associated with the raw plenoptic image(s) to be processed is defined. The raw plenoptic image directory and output file paths are also specified. The magnification at the nominal focal plane is input by the user, along with the focal length of the main lens. Other parameters such as the focal length of the microlens array are automatically loaded when specifying the plenoptic camera used in the experiment. Alternatively, the user may select the 'other camera' option and will be prompted to input these parameters.

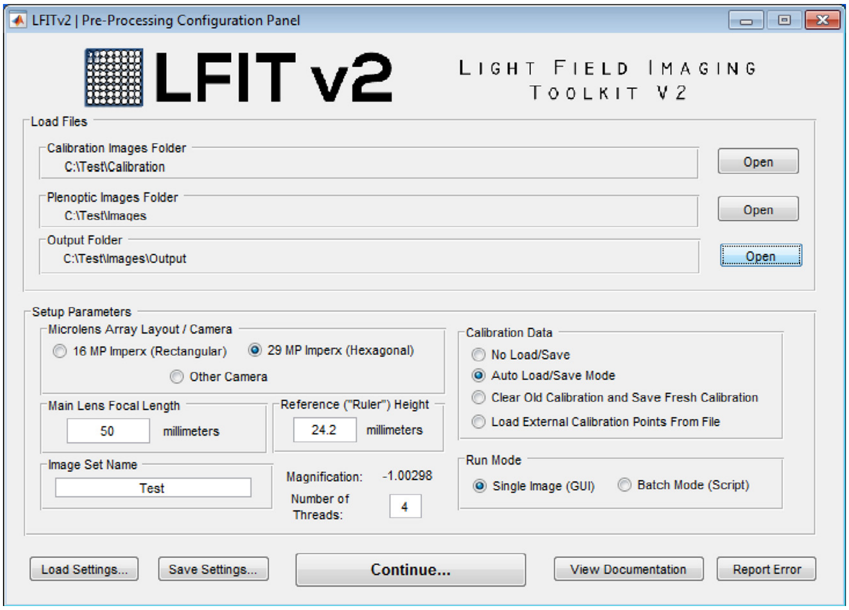
2.2. Microlens calibration

After the above parameter definitions, the first step in post-processing a plenoptic image is the microlens calibration step. To determine the center of each microimage, a series of calibration images are captured during the experiment of a white surface with the aperture of the main lens stopped down to a small size. LFIT first averages the calibration images to reduce the influence of sensor noise. After averaging, the user is prompted to identify three microlens centers in a specified pattern. The spatial relationship between these points is used to initiate a lens finding algorithm. This algorithm marches through each microimage in the scene, identifying the centroid of each spot. These centroids define the centers of each microlens.

2.3. Radiance array formation

The goal of this step is to structure the data from the raw plenoptic image into a 4D radiance array, indexed by (i, j, k, l) which correspond to (u, v, s, t) coordinates. The (u, v) coordinates are associated with the angular data, and the (s, t) coordinates are associated with the spatial data. With the center of each microimage identified via the above calibration, the pixels behind each microimage are then interpolated onto a uniform (u, v) grid because the microlens array is not perfectly aligned with the image sensor. From an algorithmic perspective, it is computationally more efficient to have the sampled (u, v) data lie on a uniform grid; otherwise, it would be later necessary to perform scattered data interpolation, which is a comparatively slow (factor of 10) process.

The known coordinates in the interpolation are derived from the calculated center of the microlens in the calibration step. In (u, v) coordinates, the microlens center will be located at $(0, 0)$. However, it is unlikely that the calculated center of the microlens perfectly aligns with a pixel in image coordinates. Because of this, the imaged pixels correspond to a sampling of slightly offset (u, v) coordinates. Thus, the known offset (u, v) values are associated with the microimage data. Using this, a new microimage can be created by interpolating for new values of (u, v) on a uniform integer-valued (not offset) grid. This process is repeated for every microimage in the raw plenoptic image, as each microimage will be non-uniformly offset relative to the center of its respective microlens. For cameras with a rectangular array, the (s, t) coordinates for the microimages are determined by assuming a non-rotated (s, t) grid centered on the center microimage. For cameras with a hexagonal array, it is necessary to resample the scattered microimage (s, t) data onto a rectilinear grid for algorithmic purposes. For both types of (u, v) interpolation, the standard MATLAB function *interpn* is used. For the hexagonal array type requiring (s, t)



Load Files

Calibration Images Folder: C:\Test\Calibration

Plenoptic Images Folder: C:\Test\Images

Output Folder: C:\Test\Images\Output

Setup Parameters

Microlens Array Layout / Camera: ○ 16 MP Imperx (Rectangular) ● 29 MP Imperx (Hexagonal) ○ Other Camera

Main Lens Focal Length: 50 millimeters

Reference ("Ruler") Height: 24.2 millimeters

Image Set Name: Test

Magnification: -1.00298

Number of Threads: 4

Calibration Data: ○ No Load/Save ● Auto Load/Save Mode ○ Clear Old Calibration and Save Fresh Calibration ○ Load External Calibration Points From File

Run Mode: ● Single Image (GUI) ○ Batch Mode (Script)

Fig. 1. The pre-processing GUI window allows a user to specify which plenoptic image to interact with, as well as load or create a microlens calibration for a particular data set.

resampling, a vectorized hexagonal-to-rectangular interpolation was employed.

2.4. Computational imaging

With the radiance array calculated, it is possible to computationally form new images from the processed data. Two types of images can be formed: perspective shifts and refocused images.

2.4.1. Perspective shifts

Perspective shifted images are an algorithmically simple type of image to form from a processed plenoptic image. Recall that each microimage consists of a series of angular samples as each microlens images the aperture of the main lens. Thus, different pixels in a microimage will correspond to different angular sets of light rays entering the camera. To create a perspective view, a (u, v) value is first chosen. A pixel corresponding to that (u, v) value is then read from every microimage in the image and stitched together to form an output image.

2.4.2. Refocusing

While perspective shifts are a natural application of the angular data captured by the plenoptic camera, perhaps the most striking application of this information is seen in the ability to computationally refocus a plenoptic image after its initial capture. That is, a single raw plenoptic image can be refocused to new arbitrary focal planes via post-processing. Two different refocusing algorithms are provided. The first (referred herein as "legacy") was derived by Ng [4].

Consider a raw plenoptic image. The camera lens (and thus the raw image) is physically focused on some nominal focal plane in the scene. It is possible to leverage the angular information recorded by the plenoptic camera to define a new focal plane at some distance offset from the nominal one. A schematic of this concept is presented below in Fig. 2. Note that A is the aperture of the main lens and $f_{mainlens}$ is the focal length of the main lens.

Consider a point source located at the nominal focal plane of the plenoptic camera and on the optical axis. The distance from this point to the main lens is the object distance, given by s_o. Light rays from such a source are indicated by the solid red line in the figure above. As seen in the diagram, these rays from the point source will converge onto the microlens array at the s plane where the point source is in-focus. The distance between the main lens and the location of the converged rays is the image distance s_i.

Now assume a point source located on the optical axis but at some depth beyond the nominal focal plane of the plenoptic camera. Rays from this point source are marked with the dashed purple line above. These rays will converge inside the plenoptic camera some distance in front of the microlens array. The rays have diverged by the time that they strike the microlens plane such that the point source is out of (nominal) focus.

By defining a new synthetic sensor plane s' at the location inside the camera where the dashed purple rays converge at s'_i, a new computationally refocused image can be formed. The previously out of focus point source at s'_o will now be in-focus in the refocused image. The depth of the refocused plane relative to the nominal focal plane is controlled by the depth parameter α. This parameter is set by the user to generate refocused images which are focused at different depths in the scene.

A second refocusing algorithm, "telecentric", is also included in LFIT. The legacy refocusing algorithm uses similar triangles to determine the size of the synthetic image sensor. Telecentric refocusing is simply a modification of the above algorithm to maintain constant magnification at each focal plane. In other words, an object translated in the depth direction

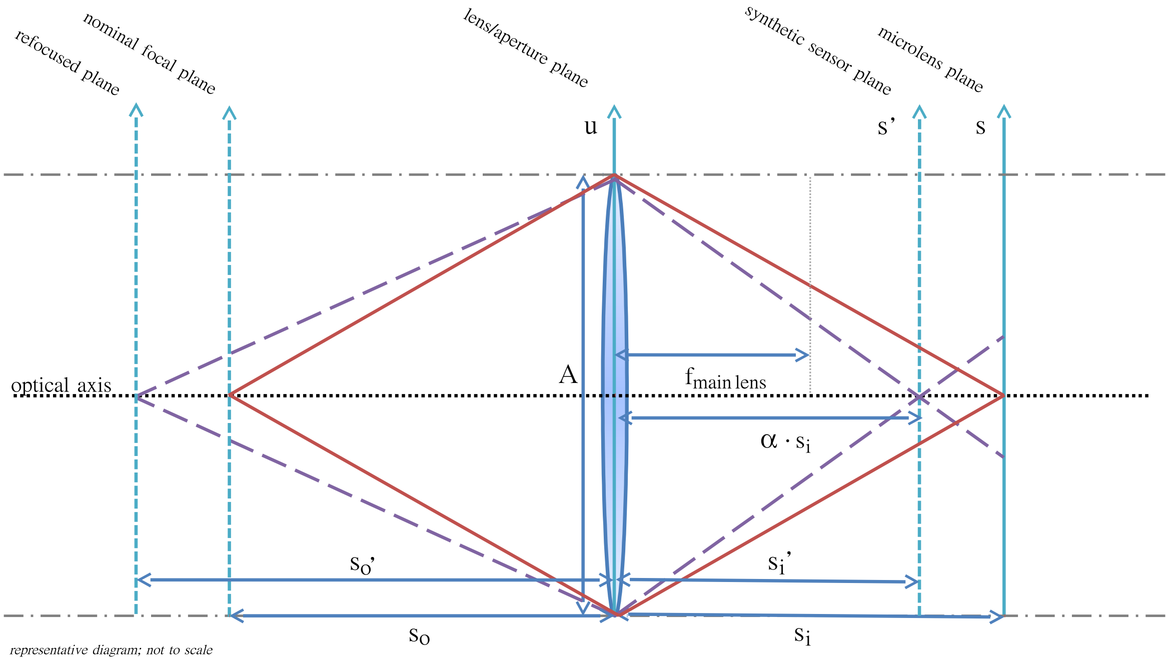


Fig. 2. Schematic depicting the process of computational refocusing. Angular information from the light field may be utilized to reconstruct an image at the synthetic sensor plane rather than the microlens plane.

will appear to be a constant size when the image is focused to the plane where it is located. In practice, it appears that some form of calibration will be needed to use this method accurately due to the effects of lens aberrations in experimental data. Work is ongoing towards developing a calibration method; however, until this is complete, quantitative determinations are discouraged with this algorithm.

3. Illustrative example

Fig. 3 is a visual flow chart depicting a typical workflow with LFIT using the GUI. A raw plenoptic image, composed of thousands of microimages, is imported into LFIT via a GUI panel along with its associated calibration image(s). Camera parameters are defined in this initial GUI along with processing options. After these settings are selected, the plenoptic image is calibrated and the radiance array is formed. A second GUI then appears for image export. Still images of arbitrary perspective views and refocused images at user-controlled depths can both be generated and saved in common image file formats.

Capability is included for forming animations of either perspective views or computational refocusing as well. For the perspective animations, the view of the scene moves about the aperture of the camera, showcasing the three-dimensionality of the data. The refocusing animations move the focal plane relative to the initial focal plane in the original scene, showing objects coming into and out of focus depending on the scene. Animations can be saved in the popular GIF animated image format or common video formats such as AVI or MP4. A sequence of refocused images, known as a focal stack, can also be generated and saved to a folder.

This second GUI panel, only present in GUI mode, allows for the loading of a new plenoptic image that is associated with the given calibration data. Note that when the main lens on the plenoptic camera is unadjusted, a given calibration image is sufficient for any number of images taken thereafter. Whenever the main lens is adjusted, a new calibration image must be acquired.

Although the above discussion dealt with the user-friendly GUI mode, a robust batch mode is also available in the demonstration script. The included demonstration script has a user-editable batch mode section, in which arbitrary code can be run in a sequential loop for all raw plenoptic images in a folder. After an initial calibration, LFIT automatically loads and processes each raw image in the folder, executing the user defined commands such as calls to the perspective or refocusing generation functions. The documentation contains extensive notes on how to properly call these functions.

4. Impact

Plenoptic cameras are increasing in popularity due to their ability to capture truly 3D data using only a single camera, which requires minimal optical access and configuration. LFIT allows any user to quickly and easily extract useful data from a plenoptic image. Together with the simplicity of the experimental setup, this will allow researchers to focus on the experimental results and new data reduction algorithms, rather than the data acquisition and processing steps. In addition, as users determine more accurate and efficient methods or implement completely new capabilities, the open source nature of LFIT will allow for rapid refinement and adoption by the research community. Within our lab, LFIT has enabled the development of 3D deconvolution [10,12,13], range finding [8,9], and background oriented Schlieren [14] using a plenoptic camera.

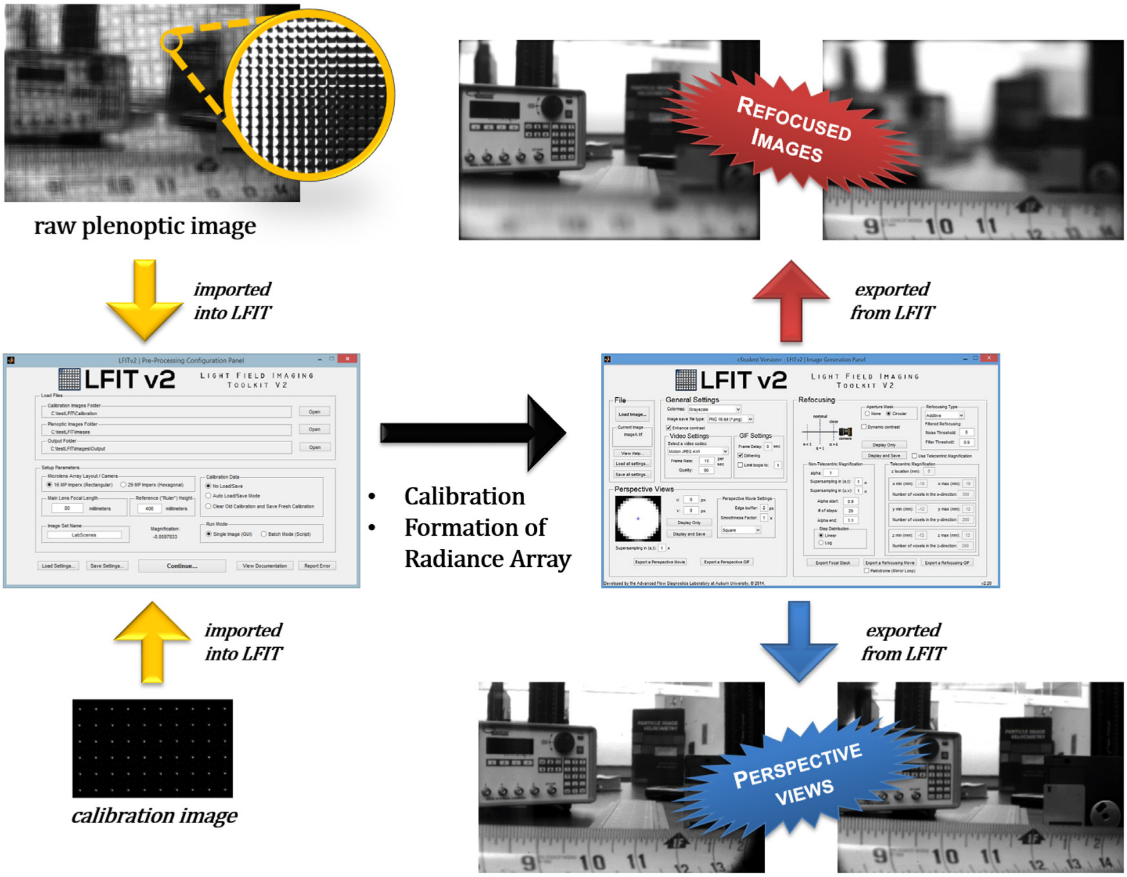


Fig. 3. Graphical flow chart depicting typical LFIT usage. A raw plenoptic image and a calibration image are used to form a radiance array, from which images at specified planes or perspectives may be reconstructed.

A number of groups utilizing LFIT include Los Alamos National Labs [14], Sandia National Labs [15], and Florida State University [16]. Additional work not yet published is ongoing at Arnold Engineering Development Center, Shanghai Jiao Tong University, NASA Langley, and others. It is the authors' hope that the existence of a well-developed and open codebase for the interactive processing of plenoptic data removes a barrier to entry and helps with the proliferation of plenoptic cameras.

5. Conclusions

Prior to the development of LFIT, a few limited implementations of the basic refocusing and perspective shift equations existed. Different authors made varying assumptions in developing algorithms, making comparisons between computed images difficult. The creation of LFIT has provided a unified foundation for our lab and our collaborators to build upon. New internal developments in areas such as 3D deconvolution, range finding, and background oriented Schlieren imaging are being facilitated by the existence of LFIT. LFIT was designed to streamline the workflow for those processing plenoptic data so that researchers could spend their time developing new applications and conducting meaningful research rather than debugging calibration algorithms or user interfaces. LFIT has been shared with various laboratories in the past but has now reached sufficient maturity that it is desired for others to benefit from our work. It is presented here as a stable release, although development will continue in order to add new features and better serve the users of this software.

References

[1] Adelson E, Bergen J. The plenoptic function and the elements of early vision. Comput Model Vis Proc 1991;3–20.
[2] Adelson EH, a Wang JY. Single lens stereo with a plenoptic camera. IEEE Trans Pattern Anal Mach Intell 1992;14(2):99–106.
[3] Levoy M, Hanrahan P. Light field rendering. In: Proc. 23rd Annu. Conf. Comput. Graph. Interact. Tech. — SIGGRAPH'96; 1996. p. 31–42.
[4] Ng R, Levoy M, Bredif M, Duval G, Horowitz M, Hanrahan P. Light field photography with a hand-held plenoptic camera. In: Stanford Tech. Rep. CTSR 2005-02; 2005.
[5] Levoy M. Light fields and computational imaging. Computer (Long Beach, CA), vol. 39, p. 46–55, 2006.
[6] Broxton M, Grosenick L, Yang S, Cohen N, Andalman A, Deisseroth K, Levoy M. Wave optics theory and 3-D deconvolution for the light field microscope. Opt Express 2013;21(21):25418–39.
[7] Thurow BS, Fahringer T. Recent development of volumetric PIV with a plenoptic camera. In: 10th Int. Symp. Part. Image Velocim.; 2013.
[8] Raynor RA. Rangefinding with a plenoptic camera. Air Force Institute of Technology; 2014.
[9] Roberts WA, Thurow BS. Correlation-based depth estimation with a plenoptic camera. In: 54th AIAA Aero. Sci. Meeting, no. 2016-0136; 2016.
[10] Bolan J. Enhancing image resolvability in obscured environments using 3D deconvolution and a plenoptic camera. Auburn University; 2015.
[11] Ng R. Digital light field photography. Stanford University; 2006.
[12] Anglin P, Reeves SJ, Thurow BS. Efficient volumetric estimation from plenoptic data. Proc. SPIE: Comp. Imag. XII 2014;9020.

[13] Anglin P, Reeves SJ, Thurow BS. Characterization of plenoptic imaging systems and efficient volumetric estimation from plenoptic data. IEEE Trans Imag Proc 2016.

[14] Klemkowsky JN, Thurow BS, Mejia-Alvarez R. 3D visualization of density gradients using a plenoptic camera and background oriented schlieren imaging, In: 54th AIAA Aerospace Sciences Meeting, no. 2016-1047; 2016.

[15] Munz ED, Guildenbecher DR, Farias PA, Thurow BS. A preliminary comparison of three dimensional particle tracking and sizing using plenoptic imaging and digital in-line holography. In: 54th AIAA Aerospace Sciences Meeting, no. 2016-1045; 2016.

[16] Bolton J, Thurow BS, Arora N, Alvi F. Volumetric measurement of a shock wave-turbulent boundary layer interaction using plenoptic particle image velocimetry. In: 32nd AIAA Aero. Meas. Tech.; 2016.