LFCamExplore: Plenoptic Camera Design Explorer

v0.1.0 in development

June 13, 2017

Homepage: http://dgd.vision/Tools/LFCamExplore/

Requires the Light Field Toolbox for MATLAB v0.4 https://goo.gl/RHe4hR.

Suggestions, bug reports, code improvements and new functionality are solicited – email Donald.Dansereau+LFToolbox at gmail dot com.

Contents

0 Quick Start

Install the light field toolbox for MATLAB, restart MATLAB, then run `LFCamExploreGUI.m`. The windows currently don't auto-arrange, so drag and resize to taste. The GUI should look like Fig. 1. Load the provided examples using the "LOAD" button, and observe changes in the spatial and sampling pattern displays as you modify the camera design and display settings.

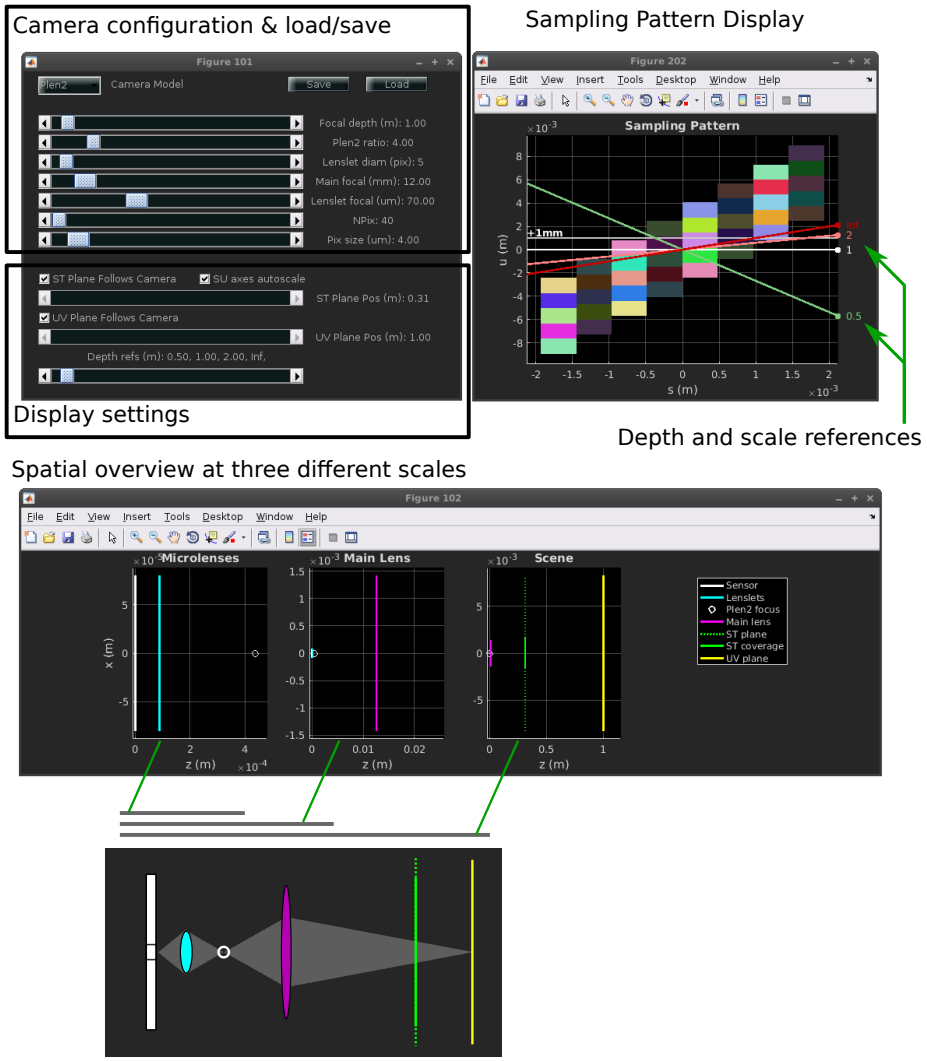


Figure 1: GUI Structure

1 Introduction

This is a tool for building intuition in the design space of plenoptic cameras. It can simulate both "focused" (aka "plenoptic 2.0") cameras as in the design by Lumsdaine and Georgiev, and "afocal", "standard" or "plenoptic 1.0" cameras following Ng's design.

The GUI is broken into three windows, as shown in Fig. 1. It accepts camera descriptions in terms of lens and sensor properties, and solves for a valid plenoptic camera based on those properties.

The key outputs are a spatial overview of the physical layout of the camera, and a display depicting the sampling pattern of the camera.

1.1 Intended Audience

This tool is intended for those wanting to design their own plenoptic cameras. The behaviours of these cameras are not always obvious, and this tool is designed as an intuition builder around the tradeoffs inherent in this design space.

For those wanting to learn more, Sect. 2 briefly touches on some of the core concepts in plenoptic imaging, and Sect. 7 points out some good references for getting started and learning more in this space.

1.2 Capabilities and Limitations

The core functionality of the program is to take a set of user-controlled parameters, solve for the geometry of a physically achievable camera fitting those parameters, and display the sampling pattern for an idealized model of that camera. An estimate of key camera performance metrics is also produced in the MATLAB terminal. Depth references are drawn to allow evaluation with respect to typical scene content.

The camera model is idealized and misses several higher-order effects. However, the program is still useful for building intuition.

The sampling pattern visualization is not responsive for large pixel counts. The intent is to visualize a small patch of pixels near the center of the sensor, to understand how the sampling pattern is influenced by key design parameters. Gross performance metrics can be calculated by editing the camera design (.json) file for a larger pixel count and loading using the "load" button, to run non-interactively.

These additional limitations apply:

- The results are entirely in 2D. Generalization to 4D should be done with care.
- This is an early release, the solvers need checking.
- The camera performance metrics can be misleading and need checking.
- This deals with paraxial ray optics only, diffraction limits and lens aberrations are not considered and will have important consequences.
- The solver and visualization only work for an integer number of pixels per lenslet. This ignores the effect of projection through the lenslet array. Vignetting is not considered.
- The sliders are somewhat coarse, and so a text entry would be useful; for now use a text editor to edit then load the .json files.

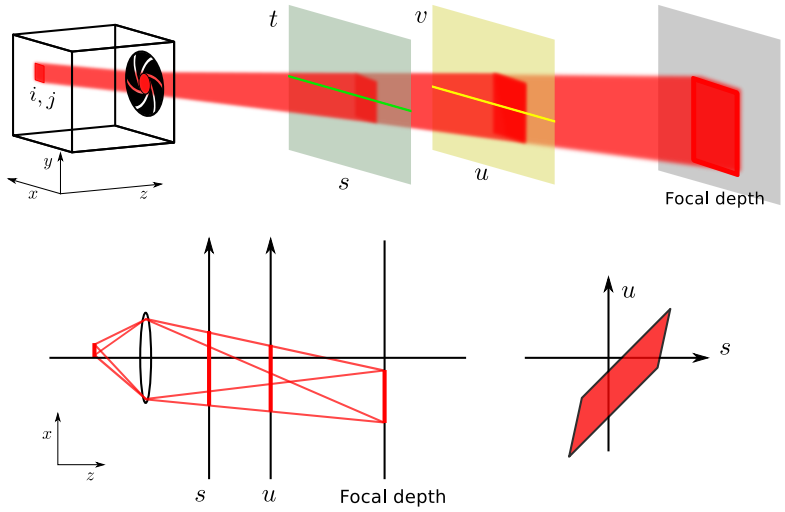


Figure 2: Sample pattern for a single pixel in a conventional camera. (top) The camera on the left is focused on the plane to the far right. The pixel at i, j integrates a subset of the light emanating from the scene, capturing some combinations of position and direction, but not others. We employ a two-plane parameterization to characterize the rays that the pixel integrates. (bottom-left) Working in 2D, we see the same situation depicted as a slice in the horizontal plane ($y = 0$). (bottom-right) By plotting the points of intersection of each ray with s and u, we obtain an epipolar image in s, u for $t, v = 0$. In this space the pixel integrates over a parallelogram.

1.3 Conventions

This program uses the absolute two-plane parameterization with the camera pointed along the z axis, as depicted in Fig. 3. Reference plane positions and therefore plane separation D are set by the user. The camera solver always keeps the sensor at $z = 0$, and all distances are measured relative to the origin, not the front of the camera – this means that a camera focused at a depth of 2 m will have its sensor at $z = 0m$, a focal plane at $z = 2m$, and the distance from the front of the main lens to the scene will be less than 2 m.

2 Understanding Sampling Patterns

In this section we review the fundamentals of sampling patterns, epipolar images, and the point-plane correspondence. Readers familiar with these concepts can skim or skip to Sect. 3.

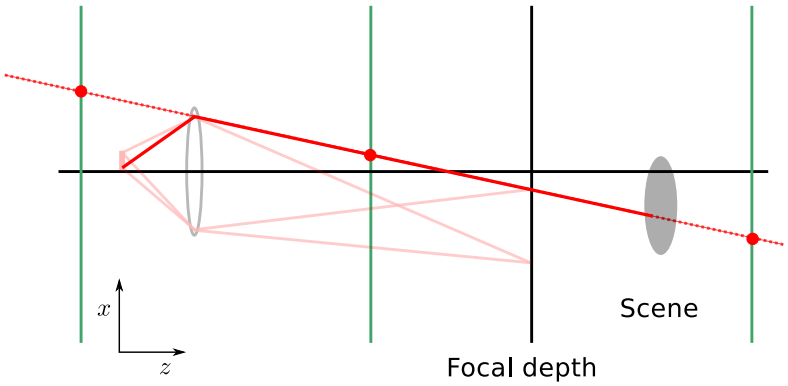


Figure 3: Rays are finite in extent, but when parameterizing we extend them to infinity. At the three potential locations for a reference plane shown in green, we see the corresponding points of intersection with the ray as it is extended to infinite length. We do not deal with the path of the ray after it has been modified by refraction or reflection – i.e. we do not consider its path inside the camera or its intersection with the scene.

2.1 Pixels as Integrators of Light

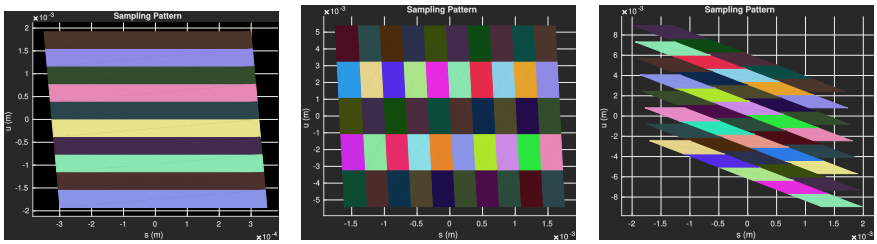
This tool characterizes a camera's performance in terms of how it samples light in the scene. All cameras map light to pixels, with each pixel integrating some subset of all the light in the scene – a subset of the so-called plenoptic function. Past work has variously characterized the sampling patterns of cameras in terms of how light behaves inside the camera, outside the camera, or a mixture of the two. Here we choose scene-space characterization in order to allow easy comparison between different camera architectures, and evaluate performance in terms of how the camera measures the world.

Referring to the top of Fig. 2, we assume a camera pointing down the z axis and centered at $x = y = 0$, with y pointing up. We assume the camera is focused at some depth, depicted on the right of the figure, and wish to describe the rays that each of this camera's pixels integrates. A single pixel at image-space index i, j is shown, and the rays that it integrates are depicted as a red volume. Note that the camera in this figure is intentionally generic, as the important details are to the right of this figure: which rays did the pixel see?

2.2 Reference Planes

One can think of a flow of light from the scene towards the camera, with only a small subset of that light eventually reaching any given pixel. To characterize this subset, we have many choices. It turns out to be convenient, in the space of plenoptic camera design, to use a light field parameterization in which each ray is described by its point of intersection with each of two reference planes.

By convention we choose the two reference planes as depicted in Fig. 2, an s, t plane and a u, v plane, with each orthogonal to the principal ray of the camera, parallel to each other, and with s and u parallel with x, and t and v parallel with y.



(a) (b) (c)

Figure 4: Example sample patterns for a horizontal group of pixels in (a) a conventional camera, (b) an standard plenoptic camera, and (c) a plenoptic 2.0 camera.

The positions of the reference planes are arbitrary. It is important is that the rays being characterized are those emanating from the scene, and not those within the camera. To accomplish this, rays are treated as lines of infinite extent, even though we know they are generally line segments with finite extent. This is depicted in Fig. 3. If a reference plane is chosen to be behind the camera relative to the scene, each ray from the scene is extended, as though the camera were not there, until it intersects the plane. Similarly if a reference plane is behind the scene, we extend the rays through the scene until they intersect the reference plane.

We will discuss in Sect. 3.2 some useful choices for the reference plane positions.

2.3 2D Epipolar Images

The reference planes give us a 4D description of the rays that each pixel integrates. This is difficult to visualize in 4D, and so we limit our attention to a horizontal slice of the scene. Referring to the bottom of Fig. 2, we set $y = 0$ and consider only each ray's s and u coordinates. We can plot all those combinations of s and u for which a ray successfully travels from the scene to reach the pixel. As depicted on the bottom-right of the figure, this results in a surface describing the integrating volume of the pixel – in this case, a parallelogram.

A plot depicting the pairs s, u or t, v as in the bottom-right of Fig. 2 is called an epipolar plot, and it turns out to be a convenient space in which to understand plenoptic cameras. From this plot one can visually evaluate complex quantities like depth of field, baseline, and spatial and angular pixel count and resolution.

2.4 How Pixels Appear

Referring again to Fig. 2, we can see that a single pixel integrates over a parallelogram in the epipolar image. This is true of both conventional and light field cameras. The four corners of the parallelogram correspond to the "marginal" rays at the edges of the pixel and the aperture. The four lines depicted in the figure correspond to these four "marginal" rays, and to the four corners of the parallelogram in the s, u plot. For lenslet-based cameras, the four corners are determined by rays at the edges of the pixels and lenslets, and not the edges of the main lens. This is the phenomenon by which depth of field is decoupled from light gathering in plenoptic cameras.

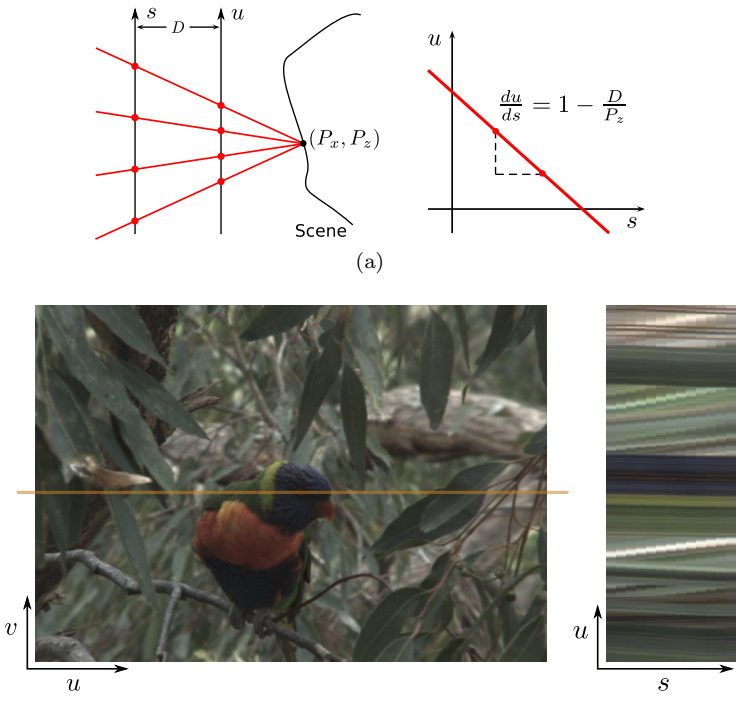


Figure 5: (a, left) A point $P = (P_x, P_y, P_z)$ in the scene emits a pencil of rays which intersect the reference planes following a linear relationship. (a, right) in the epipolar plane the point appears as a line. The position of the line depends on the 3D position of the point, but its slope is determined entirely by the depth of the point, P_z. (b, left) An example scene as a single slice in u, v, and (b, right) an epipolar slice taken at the vertical coordinate v marked by the orange line. Notice objects at different depths appearing at different slopes in the epipolar image.

If we consider multiple pixels in a conventional camera, e.g. those horizontally adjacent to the first, we can plot each in its own colour to yield an epipolar plot like the one shown in Fig. 4a. For this figure the s, t plane is at the origin, and the u, v plane is at $z = 1$ m, coincident with the focal depth of the camera. Doing the same for plenoptic cameras in the standard and 2.0 configurations yields tilings of parallelograms in the epipolar image, as seen in Figs. 4b and 4c. For the standard plenoptic camera, there are 10 pixels per lenslet, appearing horizontally along the figure, and 5 lenslets, appearing vertically. For the plenoptic 2.0 camera, there are 5 pixels per lenslet and 8 lenslets are depicted.

2.5 How the Scene Appears

So far we've used the epipolar space to describe the rays captured by each pixel of the camera, but this same space can describe the space of rays generated by the scene.

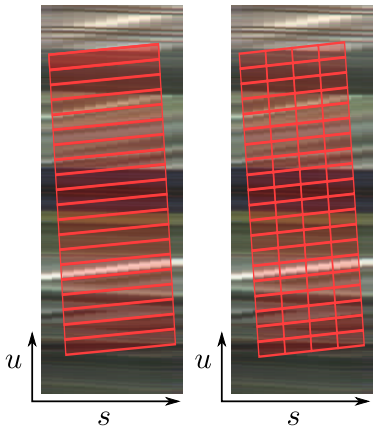


Figure 6: Two examples of the scene's epipolar image overlaid with the camera's sampling pattern: (left) for a conventional camera, and (right) for a plenoptic 1.0 camera. Both cameras are focused on a distant object, corresponding to the bright white streak in the bottom half of the images.

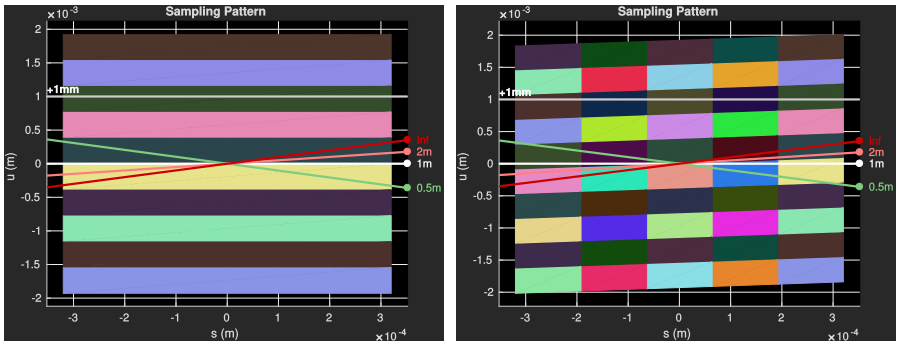
Fig. 5a demonstrates the relationship between a point in space and its appearance in the light field. For rays emanating from a point in space $P = (P_x, P_y, P_z)$, there is a linear relationship between their points of intersection with the s, t and u, v planes. This linear relationship means the point will appear in an epipolar image as a straight line, with slope related to the depth of the point P_z. This is true regardless of the surface properties at the point. If the surface happens to be Lambertian, then the line in the epipolar image will be constant-valued.

By superposition, a set of surfaces in a complex scene will appear as a set of lines in the epipolar image, with parts of the scene at similar depths appearing as lines with similar slopes. An example of this is shown in Fig. 5b. Taking a horizontal slice of the scene at the bird's eye, we obtain the epipolar image on the right, in which the bird's eye is clearly visible at a slope around 0

3 The Sampling Pattern Display

The performance of a camera for a given scene can be understood by bringing together an epipolar image for the scene, like the one shown in Fig. 5b, with the an epipolar image of the sampling pattern, like the ones shown in Fig. 4. Examples of the combined figure are shown in Fig. 6, for cameras focused on a distant object in the scene.

From this kind of figure we can determine from visual inspection which parts of the scene will appear in focus, which will exhibit defocus blur, based on how textural lines align with the pixels. The circle of confusion for a point can be evaluated visually. We can tell which structures will be large enough for the camera to resolve. The total area of each pixel tells us how much light it receives. If we were plotting all pixels for the camera, the total extent along u would tell us the camera's field of view (FOV). We get an intuition for the camera's ability to distinguish different slopes, and therefore depths in the scene.



(a) (b)

Figure 7: (a) Sampling patterns with depth references for (a) a conventional and (b) standard plenoptic camera.

The camera design GUI provides exactly this type of figure, but employing depth reference lines to emulate the content of the scene.

3.1 Depth References

In Sect. 3 we saw that superimposing the camera's sampling pattern on the scene's epipolar image yields insights into the camera's behaviour. Rather than displaying a specific scene's epipolar image, the camera design tool displays depth references like those shown in Fig. 7. These show the slopes in the epipolar image of scene elements at different depths, allowing visual inspection of the depth of field and textural resolution of the camera.

The depth references correspond to points lying along the optical axis of the camera, occupying a set of discrete depths. There is a "key" depth reference, shown in white, which is typically chosen to coincide with the focal depth of the camera. At this key depth, an additional reference is included having a 1 mm offset in the x direction. This textural reference allows us to evaluate the pixel footprint – i.e. the camera's resolution – at the key depth.

For example, Fig. 7a depicts a conventional camera. We can see that the camera is focused at 1 m from the slopes of the pixels compared to the depth references. A point at 1 or 2 m from the camera will only cross one pixel along u and will therefore appear in focus. Points at infinity or 0.5 m will cross about 2 pixels, meaning their circle of confusion will be about 2 pixels and they will appear out of focus. Numerically we confirm that the depth of field of this camera is from about 0.6 to 2.5 m.

For Fig. 7b, a standard plenoptic camera focused between 1 and 2 m, we see that objects are in focus for 0.5 m and at infinity, since points at those depths do not cross more than 1 pixel's width in the u dimension. From the textural reference at $+1mm$ we see that both these cameras have almost 3 pixels beteen the 1 m mark and the +1 mm mark, meaning the cameras will both naively have 1/3 mm resolution. The reality is actually finer for the plenoptic camera, due to the extra sampling along s allowing various forms of light field super-resolution – see Ihrke et al. [1] for more on this.

3.2 Choice of Reference Plane Locations

As our focus is on camera design, an obvious choice is to fix the u, v reference plane at a fixed depth within the scene, and the s, t plane at $z = 0$. This allows direct comparison of cameras, but can yield difficult-to-understand sampling patterns, especially for plenoptic 2.0 configurations. See, for example, Fig. 8, for which the left images show the result of this fixed approach. It is difficult to visually evaluate the performance of the plenoptic 2.0 camera on the bottom-left.

A few insights allow us to select reference plane locations to better build intuition. The standard plenoptic camera with $N \times N$ pixels per lenslet and $M \times M$ lenslets can be idealized as producing an $N \times N$ array of cameras floating at one focal distance in front of the main lens, with each camera having $M \times M$ pixels.

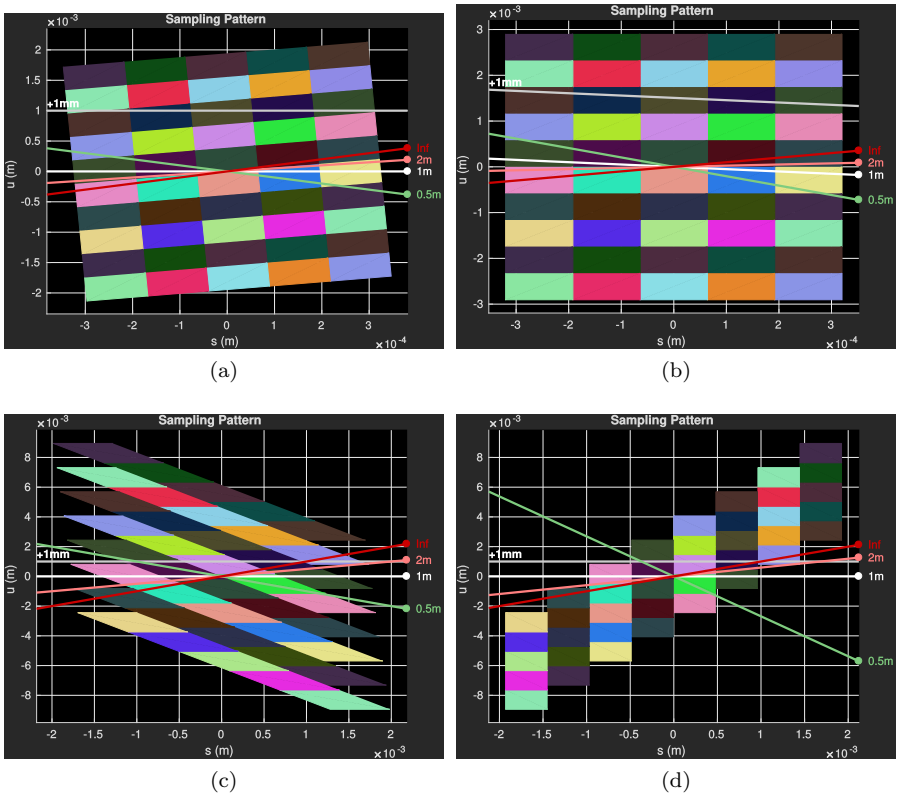


Figure 8: Choice of reference plane positions for (top) a standard plenoptic camera and (bottom) a plenoptic 2.0 camera: (left) fixing the s, t plane at $z = 0$ and the u, v plane at 1 m, compared with (right) placing the s, t plane at the virtual camera array position (see main text), and the u, v plane at the camera's focal depth. The two scenarios convey the same information but in different forms.

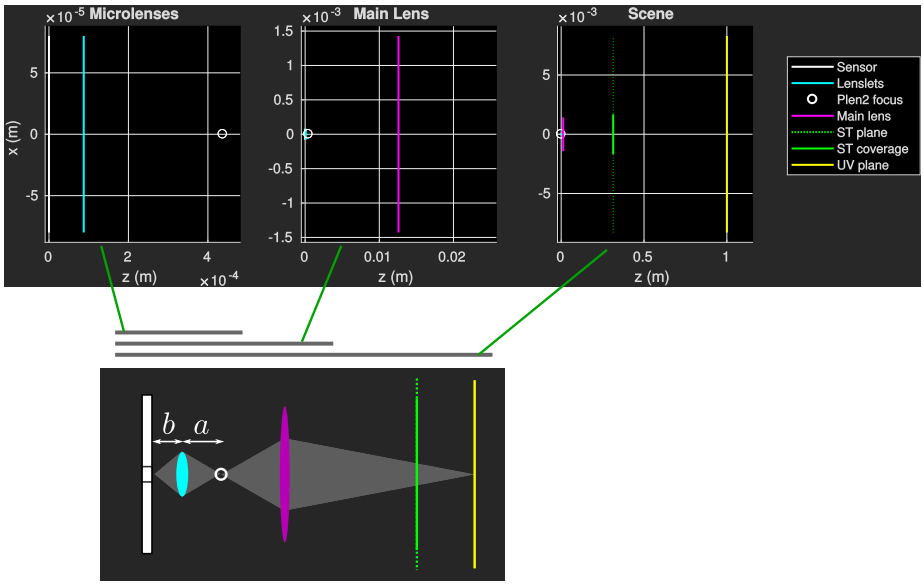


Figure 9: The spatial overview display gives a rough depiction of the positioning between the sensor, lenslet array, main lens, reference planes and, for plenoptic 2.0 cameras, the plenoptic focal point. Depicted here are the three scales employed to convey the entire scenario.

Similarly, the plenoptic 2.0 camera with the same lenslet and pixel count can be understood as producing an array of $M \times M$ cameras inside the scene, each having $N \times N$ pixels. The position of the array depends on the focal settings of the camera, and is typically between the scene and camera, but can also be behind the scene or camera.

So both camera configurations can be thought of as virtual camera arrays. If we fix the s, t plane at this virtual camera array position, and the the u, v plane at the focal depth of the camera, we can obtain better aligned sampling patterns. This is demonstrated on the right in Fig. 8. Note that the depth references reflect the changes to reference plane positions, so that these figures depict the exact same information, just in a different form.

The plenoptic 2.0 example in Fig. 8 is paticularly interesting, as it is difficult to tell from the version on the left that most scene points are only seen 4 times, while this is clear by following the 1 m depth reference through the sampling pattern in the figure on the right. It is also clear from this plot that the resolution behaviour of the plenoptic 2.0 camera is much more complex, yielding differing sample counts along s based on depth. Note that for this camera an object at infinity will be seen 6 times, while one at 0.5 m only about 2 times.

Table 1: GUI-controlled display options

Setting	Description
s, t plane follows...	Controls the s, t plane to the virtual camera array depth, as in Sect. 3.2
u, v plane follows...	Controls the u, v plane to the focal depth, as in Sect. 3.2
s, t and u, v plane positions	Manual control of the reference plane positions
Depth refs	Enable and control the key depth of the depth references. Displayed are the key depth times 0.5, 1, 2 and infinity
s, u axes autoscale	Enable to have the epipolar plot resize to fit the sampling pattern

Table 2: GUI-controlled camera parameters

Setting	Description
Camera model	Selects a plenoptic 1.0 or 2.0 configuration
Focal depth (m)	Depth at which the camera's main lens is focused
Plen2 ratio	Ratio of the focal distances a/b, see Fig. 9; only for plenoptic 2.0 cameras; negative means focus behind the lenslet array
Lenslet diam (pix)	Size of a single lenslet as an integer number of pixels
Main focal (mm)	Focal length of the main lens
Lenslet focal (µm)	Focal length of the microlenses
Pixel count	Number of pixels to simulate
Pixel size (µm)	Size of a single pixel

4 The Spatial Overview Display

The spatial overview depicts the physical layout of the camera, in a set of rough plots, to help build intuition and reality check a design. Because the relevant dimensions span from micrometers to meters, this cannot be done to scale in one figure. The spatial layout is therefore split between three figures, one at the scale of the microlenses, one at the scale of the main lens, and one at the scale of the scene. Fig. 9 depicts three typical figures for a plenoptic 2.0 camera, and how they relate to a not-to-scale model of a plenoptic 2.0 camera. Colour coding is maintained through the sub-plots and reference diagram, and follows the legend to the right of the figure.

Table 3: Computed camera parameters

Setting	Description
SensorSize (m)	Total extent of the sensor
LensletDiam (m)	Diameter of a single lenslet, in meters
NLenslets	Total lenslet count
MainLensDiam (m)	Diameter of the main lens
MainLensDist (m)	Position of the main lens relative to the sensor
LensletDist (m)	Position of the microlens array relative to the sensor
Plen2FocDist (m)	Distance of the plenoptic 2.0 focal point relative to the sensor

5 Explanation of Parameters

Table 2 summarizes the camera design parameters controlled through the GUI. The solver uses these values to compute the camera parameters summarized in Table 3. The computed parameters currently appear in the MATLAB console, and some of them are depicted in the spatial overview figure.

A set of camera performance metrics are also estimated, and these appear in the MATLAB console. These are summarized in Table 4. Use these with caution, they are experimental and can be misleading, e.g. the number of angular samples does not map trivialy to depth resolution.

6 Future Plans

The experimental performance estimate is the weakest feature. It should be upgraded to include depth resolution, a spatial resolution that takes the light field super-resolution into consideration, and ideally depth-dependent spatial and angular resolution plots.

The spatial displays are very minimalist, some way of producing a nicer display without impacting the update rate would be ideal.

It would be better not to reset the display options on load.

This document would benefit from a section exploring practical examples.

7 Further Reading

The light field parameterization was introduced in [2] and an excellent review of the light field structure, sampling patterns in world space, and pointers to further reading are given by Ihrke et al in [1].

Light field basics, parameterizations and visualization are discussed in [3] Chap. 2, pixel integrating volumes and sampling patterns in Chap. 3.2, and the point-plane correspondence in Chap. 4.3.

Table 4: Computed performance estimate

Setting	Description
Theoretic FOV (deg)	Field of view; typically the tool would be used with a subset of the total pixel count, but if set to the full horizontal pixel count this will reflect the camera's FOV
FOV Footprint (mm)	Footprint (captured area) of the image at the focal depth, see pixel count comment above
Naive spatial samps	Number of samples in the u direction at the focal depth, i.e. "spatial" samples, with respect to the scene
PixelFootprint (mm)	Footprint (integrated area) for a pixel at the focal depth
MaxLensletAlignError (m)	Maximum alignment error between the lenslet array and sensor planes before a pixel-sized shift is observed; this is also reported in deg. and arcsec
EffectiveFNumber	A number that reflects the light gathering ability of the camera: distance from main lens to lenslet array, divided by main lens diameter
LensletFNumber	F number of the lenslets
MainLensFNumber	F number of the main lens
NumAngularSamples	Number of samples in the s direction at the focal depth, i.e. "angular" samples, with respect to the scene
NaiveNearFocalLimit (m)	Near focal limit
NaiveFarFocalLimit (m)	Far focal limit

Plenoptic camera design, camera-space sampling patterns and epipolar images are discussed in [4] Chaps. 2 and 3, and the focused (plenoptic 2.0) configuration is introduced in [5].

References

[1] I. Ihrke, J. Restrepo, and L. Mignard-Debise, "Principles of light field imaging," *IEEE Signal Processing Magazine*, vol. 1053, no. 5888/16, 2016. 10, 14

[2] M. Levoy and P. Hanrahan, "Light field rendering," in *SIGGRAPH*. ACM, 1996, pp. 31–42. 14

[3] D. G. Dansereau, "Plenoptic signal processing for robust vision in field robotics," Ph.D. dissertation, Australian Centre for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Jan. 2014. [Online]. Available: http://www-personal.acfr.usyd.edu.au/ddan1654/Dansereau2014Plenoptic.pdf 14

[4] R. Ng, "Digital light field photography," Ph.D. dissertation, Stanford University, 2006. 15

[5] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *Computational Photography (ICCP)*. IEEE, 2009, pp. 1–8. 15