Algorithms for microlens center detection

Maximilian Schambach and Fernando Puente León

Karlsruhe Institute of Technology
Institute of Industrial Information Technology
Hertzstr. 16, 76187 Karlsruhe

Abstract We investigate four algorithms for microlens center detection, two of which have not been previously discussed in the literature. Using a physical approach, we create a set of 81 synthetic white images with known microlens center coordinates. Applying the different detection algorithms to the synthetic white images, we are able to quantitatively evaluate their respective performance in terms of accuracy, precision and recall. Overall, the proposed methods outperform the ones that have been previously published.

Keywords Image processing, computational imaging, microlens array.

1 Introduction

In the scope of geometrical optics, light is described as rays whose optical path lengths, according to Fermat's principle, attain an extremum. Geometrical optics is well suited to describe the imaging process of cameras but as such is lacking the description of many essential properties of light. By means of an extension to geometrical optics, properties such as color and intensity, which can only be described within wave optics (or higher order theories such as quantum electrodynamics), can be heuristically incorporated into geometrical optics: The *light field* (LF) or *plenoptic function* $L_{\lambda,t}(x, y, z, \phi, \theta)$ describes the optical radiance at point (x, y, z) in direction (ϕ, θ) of wavelength λ at time t in units $\mathrm{W m^{-2} sr^{-1}}$. In homogeneous media that are free of occluders, the radiance along a ray is constant. The spatial dependency of the LF can hence be reduced by one dimension, resulting in the so-called *4D light field* $L_{\lambda,t}(u, v, a, b)$, where the coordinates (u, v, a, b) correspond to a certain parametrization of the spatial dependency of the LF of which there are numerous. For LF cameras, one usually uses the plane-plane parametrization: A light ray inside a camera is uniquely described by the intersection points (u, v) and (a, b) of two parallel planes, the main lens plane and the sensor plane.

At every sensor coordinate (a, b), conventional cameras perform an integration of the LF over the time t, the wavelength λ as well as the coordinates (u, v) which encode the direction of the incident light ray. The aim of LF cameras is to measure this angular dependency that is lost in conventional camera systems. LF cameras have been part of active research over the past two decades and open new possibilities to measure spatial information, in particular the depth, of a scene. There are different implementations of LF cameras such as camera arrays, gantry or microlens array (MLA) based LF cameras. Camera arrays and gantry based LF cameras are the optically most straightforward way to construct a LF camera. They offer very good spatial and angular resolution but require sophisticated calibration schemes, are bulky and mechanically sensitive. MLA based cameras [1,2] have recently gained in popularity due to their compact monocular design. By placing a MLA in front of the imaging sensor, the direction (u, v) of incident light rays is coded into the sensor image: Light rays hitting the sensor plane perpendicularly will be imaged onto the central pixel underneath a microlens (ML), slanted rays will be imaged onto pixels deviating from the center. Thus, the relative position of the image point w. r. t. the ML center codes the (u, v) coordinate of the incident light field, the position of the ML itself the (a, b) coordinate.

The foundation of all calibration and decoding schemes needed in the case of MLA based LF cameras is the exact detection of the ML centers with subpixel precision. The ML centers are detected using so-called *white images* (WIs) – images of a white scene for example taken using an optical diffuser. The calibration and decoding quality is dependent on the accuracy and robustness of the ML center detection. In spite of the importance of the ML center detection, to our knowledge there is no literature investigating the quality of the ML center detection algorithms. This is the main scope of this article. The results presented in this paper analogously apply for MLA based computational cameras other than LF cameras with little or no adjustment, e. g. MLA based spectral cameras.

The paper is organized as follows. Following this short introduction, we outline two existing and propose two new ML center detection algorithms in Section 2. In Section 3, we first present the methods used to create reference data and the used evaluation metrics. Using the synthetic ground truth data, we present a performance evaluation of the different ML center detection algorithms. Section 4 concludes the paper with a brief summary.

2 Microlens center detection

Challenges in the detection of MLs and their centers are versatile: On the one hand, the sheer amount of MLs in MLAs used in practice (in the case of the Lytro Illum camera about 150.000 MLs) limits the algorithm's complexity. On the other hand, the geometry of the MLA is not trivial and usually slightly irregular: Ideally, the MLs are circular, arranged in a perfect rectangular or hexagonal grid and perfectly aligned with the sensor. In practice, the MLA will be translated, rotated and scaled[1] w. r. t. the sensor and the lattice will be slightly irregular due to manufacturing tolerances. Furthermore, main lens and ML vignetting influences the form and brightness of the ML images, particularly of those that are close to the sensor edge.

There are two ML detection methods proposed in the literature: Cho et al. [3] first perform a greyscale erosion and clustering of the WI. To estimate the ML centers, they use a parabolic least squares (LS) regression of the clustered MLs. Dansereau et al. [4] propose a decoding pipeline for the Lytro light field camera implemented, as the de-facto non-proprietary standard, in the MATLAB *Light Field Toolbox*. As a preprocessing step, the WI is convolved with a filter kernel. In the case of the MATLAB *Light Field Toolbox*, a disk kernel with a fixed radius of 1/3 of the expected ML spacing is used. The ML centers are then estimated by finding the local maxima in the filtered image[2]. We propose two detection algorithms that join parts of the aforementioned detection pipelines: First, to reduce noise and other high frequencies, we convolve the image with a filter kernel. We investigate different kernel types and

[1] That is, ML centers will not be imaged onto pixel centers.

[2] This does not result in subpixel precision. However, in the succeeding decoding pipeline of the toolbox, a ML grid model is built with subpixel precision.

sizes. In a second step, the images are clustered using local thresholding (by local Gaussian weighted mean with a block size of 17 px) to find areas around local peaks and a standard cluster labeling algorithm. Each cluster represents exactly one ML. The thresholding has to be performed locally, as the main lens vignetting results in different local maximum values for the MLs close to the sensor edge. Finally, we either perform a parabolic LS estimation to obtain the ML centers or calculate the center of mass (CoM) of the individual clusters. Both methods yield ML centers with subpixel precision but perform differently in terms of accuracy and runtime. The different ML center detection pipelines are depicted in Figure 1.

3 Evaluation

3.1 Reference data

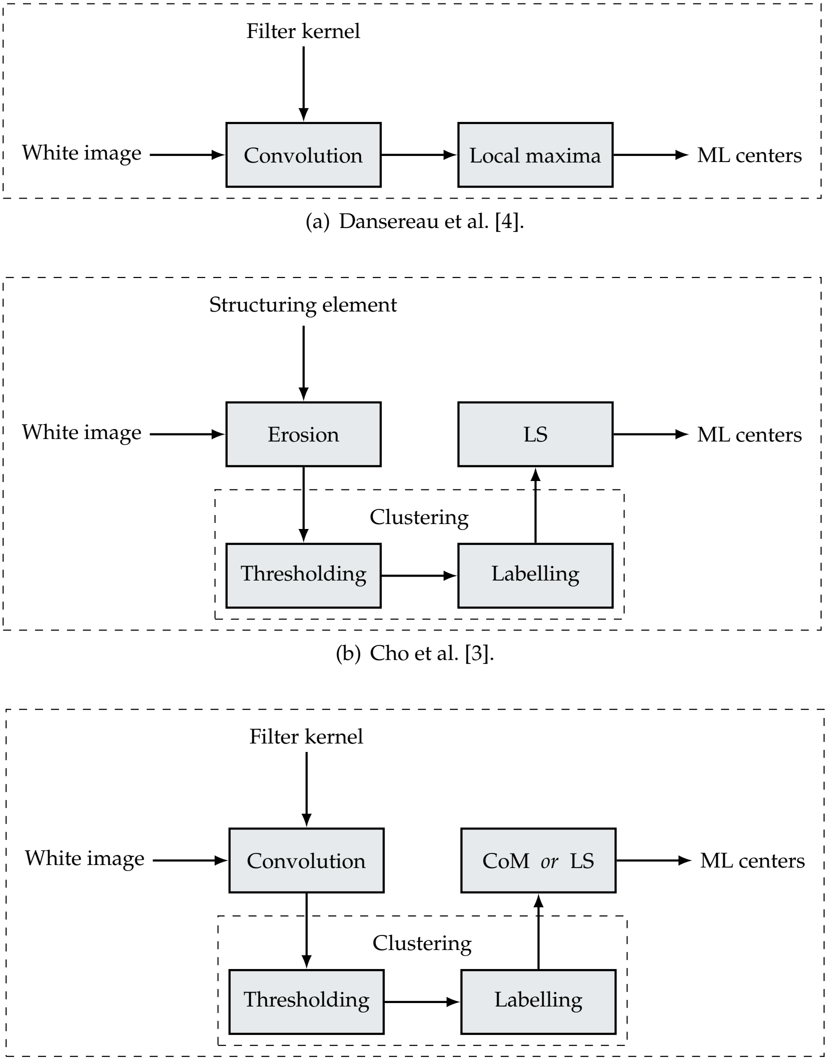
In order to objectively evaluate the performance of the ML detection algorithms, appropriate reference data is needed. Of course, real WIs, as for example provided by the Lytro cameras, are unsuited since the actual ML centers are unknown. Therefore, the reference data has to be synthesized: In general, the irradiance at a sensor coordinate (a, b) for a camera with one lens of radius r at distance d to the sensor is given by [2]

$$E(a,b) = \frac{1}{d^2} \int_0^T \int_\Lambda \iint_{\mathcal{A}} L_{\lambda,t}(u,v,a,b) \cos^4 \phi \; \mathrm{d}u \, \mathrm{d}v \, \mathrm{d}\lambda \, \mathrm{d}t \, .$$

Here, $\mathcal{A} = \left\{ \mathbf{u} = (u,v) \in \mathbb{R}^2 : \|\mathbf{u}\| < r \right\}$, T denotes the exposure time, Λ a wavelength interval and $\phi = \arctan\left(d^{-1} \cdot \left|\sqrt{u^2+v^2} - \sqrt{a^2+b^2}\right|\right)$ the angle between the incident light field and the sensor normal. For WIs, we assume that the incident light field is constant inside the camera at the MLA plane, hence the time and wavelength integration yields a constant factor. We obtain

$$E(a,b) \propto \frac{1}{2d^2} \left(\frac{A_-}{A_-^2 + 1} - \frac{A_+}{A_+^2 + 1} + \arctan A_- - \arctan A_+ \right) \, , \quad (1)$$

where $A_\pm = d^{-1} \left(\sqrt{a^2 + b^2} \pm r \right)$. Using (1), we calculate the irradiance underneath each ML. These ML illuminations are then arranged

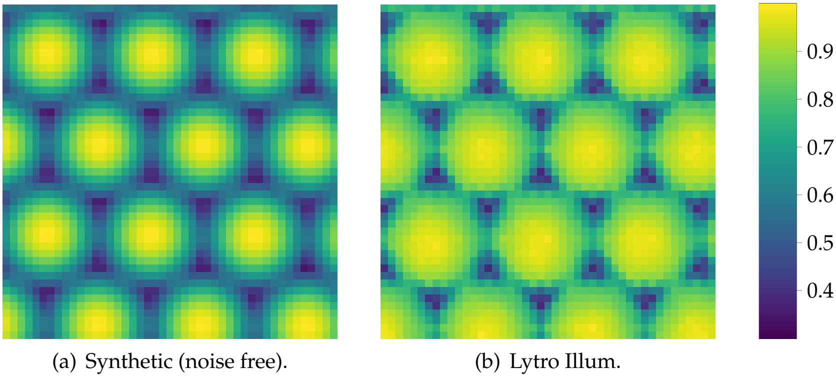


(a) Dansereau et al. [4].

(b) Cho et al. [3].

(c) Proposed (CoM) and (LS).

Figure 1: ML center detection pipelines.



(a) Synthetic (noise free).

(b) Lytro Illum.

Figure 2: Image sections of normalized WIs with ML radius $r = 7\,\mathrm{px}$. The shown Lytro WI is the mean of all 34 Lytro WIs.

in a hexagonal grid to compose a WI. Main lens vignetting is added to the WI in an analogous fashion. To simulate irregularities of the grid, we add bivariate Gaussian noise of different variances to the ideal grid point coordinates. Furthermore, we add different levels of Gaussian image noise to the WIs. Finally, the synthesized images are downsampled to 16 bit. The used parameters (ML and main lens radius, focus distance, pixel pitch, grid spacing) are taken from a metadata file of a Lytro Illum camera. The remaining parameters, such as the grid rotation and subpixel offset, are varied throughout the investigation. A comparison of a synthesized and a Lytro WI are shown in Figure 2.

Each Lytro Illum camera provides 34 white images for calibration, taken at different focus and zoom settings. These white images differ mostly in the severity of vignetting: With different focal lengths of the main lens but with unchanged entry pupil, the ML images will be crescent shaped instead of circular at the sensor edges. To be able to detect the true geometric ML centers, a preprocessing of the white images is necessary, for example by averaging all available white images. This shall not be part of this paper, as we focus on the actual ML center detection.

Using the synthetic WIs, the performance of the ML center detection algorithms can be investigated: The coordinates of the true ML centers are known and can be compared with the estimated ones. We investigate a total of 81 different synthesized WIs of size 1200 × 1800 px containing around 10 000 MLs each.

3.2 Metrics

For every WI, we calculate the estimated ML centers using the four different ML center detection pipelines for varying parameters such as the convolution filter kernel and kernel radius[3] or the size of the structuring element. For every detected center, using the ground truth data, we calculate the distance to the closest known grid point. If the distance is larger than 4 px, the detection is dismissed and regarded as a false positive. Otherwise, the measured distances are collected and the mean and variance of these measured distances are saved along with the corresponding WI and algorithm parameters. If two or more detections occur within a 4 px radius around the same grid point, only the closest one is considered a true positive. We then can calculate the accuracy Q and its standard deviation as the mean and standard deviation of all measured absolute distances of detected to actual ML centers, the precision P, the recall R and the F-measure F,

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \,, \quad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \,, \quad F = 2\frac{P \cdot R}{P + R} \,,$$

where TP (FP), FN denote the detected true (false) positives and false negatives respectively.

3.3 Results

To be able to compare the performances of every detection pipeline, we first determine the free parameters of each method with the highest detection accuracy. For this, we investigate the mean accuracy over all WIs for different sizes of the kernel respectively the structuring element. The results are shown in Figure 3.

The accuracy in case of the method proposed by Dansereau et al. shows only little dependence on the size of the used filter kernel as well

[3] We only investigate kernels of odd size with size $= 2 \cdot \text{radius} + 1$.

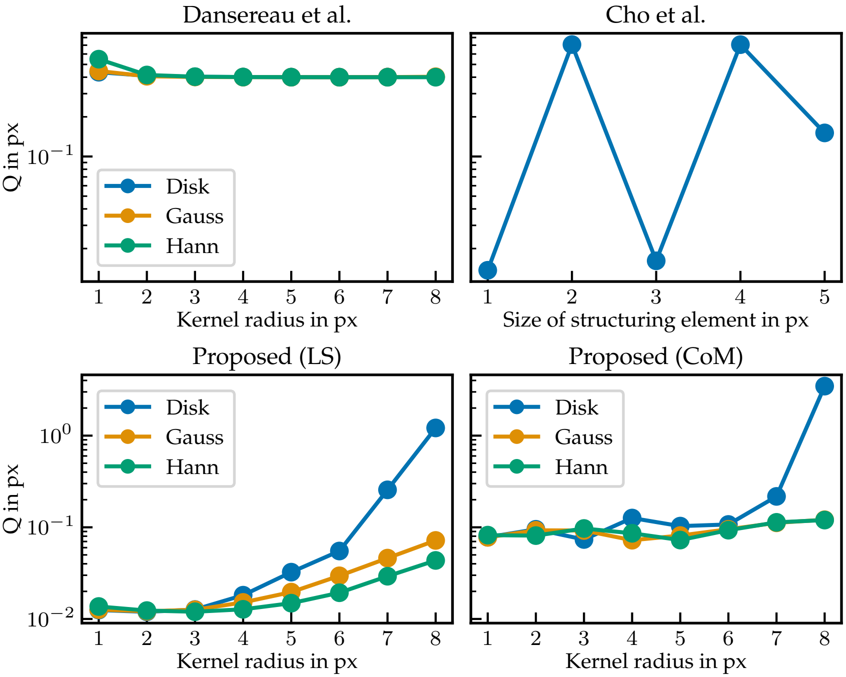


Figure 3: Meausred accuracy Q for different kernel radii respectively different sizes of the structuring element for the various detection pipelines.

as the kernel shape. Except for a kernel radius of 1 px, the algorithm performs with an mean accuracy $Q \approx 0.4$ px with a minimum at a kernel radius of 6 px. The algorithm proposed by Cho et al. shows alternating performance depending on the size of the structuring element. A minimum of $Q = 0.0137$ px is reached for a size of 1 px, corresponding to no erosion performed at all. For even-sized structuring elements, the algorithm performs significantly worse than for odd-sized ones. This is caused by shift of the picture due to the fact that even-sized structuring elements do not possess a central pixel: The shift causes a systematic error in the estimation of the ML centers. Overall, we conclude that erosion is not well suited as part of the image preprocessing in the case of

ML center detection. The proposed method using a parabolic LS estimator shows a dependence on the kernel size similar for all kernel types. A minimum is reached for kernel sizes larger than one, but the algorithm performs poorly for large kernel sizes. For large kernel sizes, the disk kernel performs the poorest. This is unsurprising as the disk kernel is not suited as a high frequency filter as it contains high frequencies itself. Overall, an optimum of $Q = 0.0120\,\mathrm{px}$ is reached for a disk kernel with 2 px radius, but similar results are obtained also with the Gauss and Hann kernel with radii of 2 and 3 px. Finally, for the proposed method using a CoM calculation, only slight dependence on the kernel size is observed, with an exception again being a large disk kernel. A minimum is obtained for a Gauss kernel of radius 4 px with $Q = 0.0723\,\mathrm{px}$. For further comparison, we choose the two best performing parameters for every algorithm.

Using these best performing parameters, detailed performance results are shown in Figure 4. Here, each scatter point represents the result calculated from one WI. Overall, the results cluster in three groups: The pipeline proposed by Dansereau et al. runs fast but has a poor performance w. r. t. the detection accuracy. The method by Cho et al. and the proposed LS method result in very accurate measurements with small standard deviations but with significantly longer runtime. Last, the proposed CoM method performs very fast with reasonable accuracy below 0.1 px. All methods reach a very high F-measure in general. Looking more closely at the recall and precision, we observe that the LS-based methods perform poorly w. r. t. precision compared to the remaining methods. The problem is inherent to the used non-linear LS estimator: Depending on the detected ML clusters, the used optimization algorithm might not converge within the desired tolerance or a given maximum number of iterations yielding an invalid ML center resulting in a FP and thus in a suboptimal precision value[4]. The proposed CoM based method and the one proposed by Dansereau et al. always perform with a precision of exactly one, meaning that no false positives are detected. This can be of importance for applications where the centers are directly used for further calculations. All methods show reasonable recall performance. Further analysis showed that suboptimal recall val-

[4] Using a linear LS estimator showed even worse numerical stability resulting in a lower precision performance with only a slight improvement of runtime.

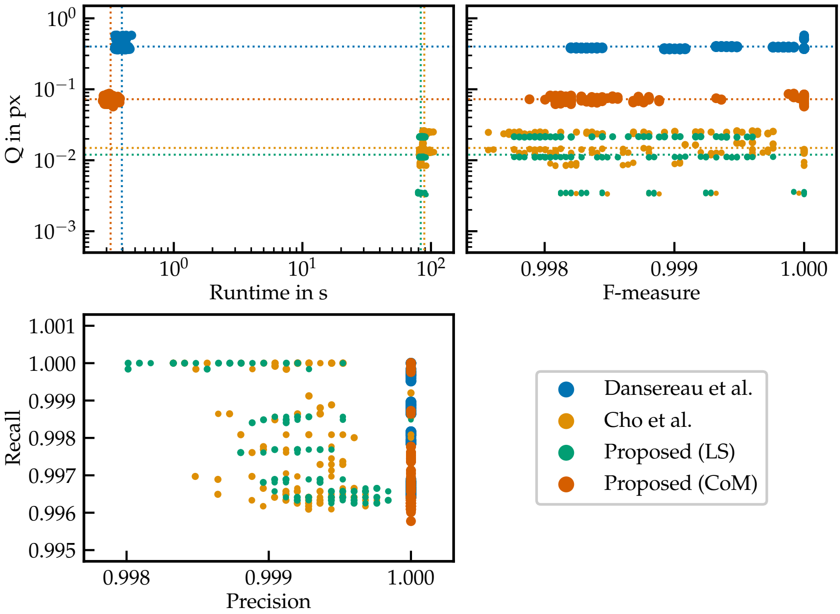


Figure 4: Overall performance comparison of the different detection pipelines. The size of each scatter point is proportional to its measurement's standard deviation. The shown lines correspond to each dataset's mean value.

ues, meaning that not all ML are detected, occur exclusively at image edges. Due to cut off ML images, in real applications, one would neglect the image edges anyway.

Finally, we have a closer look at the performance dependence on the WI properties such as rotation, grid and image noise. The results are shown in Figure 5. Overall, we again observe that the LS methods are the most accurate, with the proposed method performing slightly better than the one by Cho et al. [3]. The method by Dansereau performs worst and the proposed CoM method's performance lies in between. For grid rotation and grid noise, there is no influence on the detection accuracy for either one of the pipelines. This is not a surprise since the detection pipelines are all operating locally and every ML is detected and evaluated separately. Rotation and grid noise do not have a systematic influ-

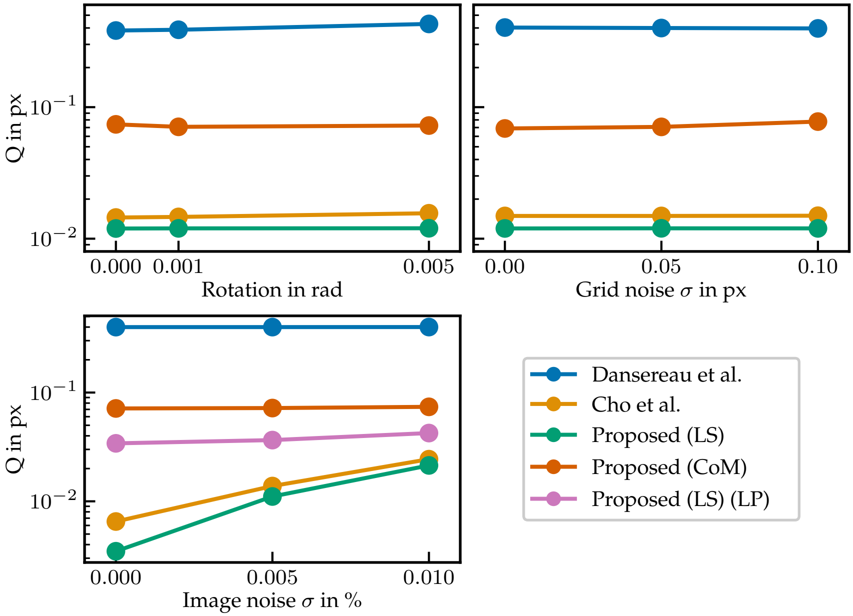


Figure 5: Performace comparison of the different detection pipelines for different WI properties.

ence on the local ML image. Image noise, on the other hand, does have a local impact on each ML image. We observe a significant influence on the LS based methods. Both the proposed LS and the method by Cho et al. perform considerably worse with increasing image noise while the methods by Dansereau et al. and the proposed CoM method are invariant to image noise. The latter perform a low-pass filtering with kernel radii of about 7 px whereas the former either perform no filtering (Cho et al.) or use very small filters (proposed LS method). Increasing the filter radius, and hence decreasing the cutoff frequency, for the proposed LS method to 7 px (depicted in Figure 5 by the (LP) label) confirms that the filtering successfully suppresses image noise. In doing so, the accuracy of the proposed LS pipeline drops significantly, resulting in an accuracy close to the proposed CoM method.

4 Conclusions

We have investigated four ML center detection pipelines, two of which have not been previously discussed in the literature. Overall, the proposed methods perform better than those having previously been proposed in the literature: The method proposed by Dansereau et al. [4] shows a fast runtime, high precision, but poor accuracy. The proposed CoM based method outperforms the previous in terms of accuracy and runtime while performing similarly in terms of precision and recall. On the other hand, the method proposed by Cho et al. [3] performs with a high accuracy but long runtimes and suboptimal precision. The proposed LS based method again outperforms the former in terms of accuracy while performing similarly in the remaining metrics. With respect to WI parameters and noise, the method by Dansereau et al. and the proposed CoM method are the most robust. Concluding, the proposed CoM method is well suited for most applications.

References

1. E. H. Adelson and J. Y. Wang, "Single lens stereo with a plenoptic camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 99–106, 1992.
2. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report*, vol. 2, no. 11, pp. 1–11, 2005.
3. D. Cho, M. Lee, S. Kim, and Y.-W. Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3280–3287.
4. D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1027–1034.