



Joint electromagnetic and ray-tracing simulations for quad-pixel sensor and computational imaging

Guillaume Chataignier, Benoit Vandame, Jérôme Vaillant

► To cite this version:

Guillaume Chataignier, Benoit Vandame, Jérôme Vaillant. Joint electromagnetic and ray-tracing simulations for quad-pixel sensor and computational imaging. Optics Express, 2019, 27 (21), pp.30486. 10.1364/OE.27.030486 . hal-04241630

HAL Id: hal-04241630

<https://hal.science/hal-04241630>

Submitted on 13 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Joint electromagnetic and ray-tracing simulations for quad-pixel sensor and computational imaging

GUILLAUME CHATAIGNIER,^{1,2,*} BENOIT VANDAME,¹ AND JÉRÔME VAILLANT²

¹InterDigital, 975 Avenue des Champs Blancs, Cesson-Sévigné, 35576, France

²Univ. Grenoble Alpes, CEA, LETI, DOPT, LIS F-38000 Grenoble

*guillaume.chataignier@interdigital.com

Abstract: Since Canon released the first dual-pixel autofocus in 2013, this technique has been used in many cameras and smartphones. Quad-pixel sensors, where a microlens covers 2x2 sub-pixels, will be the next development. In this paper we describe the design for such sensors; related wave optics simulations; and results, especially in terms of angular response. Then we propose a new method for mixing wave optics simulations with ray tracing simulations in order to generate physically accurate synthetic images. Those images are useful in a co-design approach by linking the pixel architecture, the main lens design and the computer vision algorithms.

© 2023 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

In recent years, a new tendency is observed in sensor design which exploits the capability of very high pixel counts. While the pixel density is continuously increasing following the wafer-level improvements, images extracted from sensors are down-sampled by 2 or 4 before being rendered. The break in resolution between the captured pixels and the rendered pixels gives rise to new features where pixels serve algorithms (*i.e. computational photography*). The pixel count is nowadays so high that despite the resolution break, rendered images still have a high pixel count competing with common sensors. From existing sensors one categorises 2 major usages: trivial demosaic and live auto-focus.

A first trend encompasses sensors with very high pixel counts like the Sony sensor IMX586 made of 48 MegaPixels with a 0.8 μm pixel size. The extracted image could be under-sampled by a factor of 2 in both directions allowing the under-sampled image to be trivially demosaiced. This trend appears with new smart-phones (Honor View 20, Huawei P30, or Xiaomi Redmi N7) which deliver high resolution images with very good color discrimination. Alternatively, a second trend refers to so-called *dual-pixel* sensors where each microlens covers 2 rectangular shaped sub-pixels, each one having its own photodiode. Each one observes the light entering either from the right- or left- half-pupil of the main-lens. Left & right images are obtained by splitting both pixels in 2 distinct images. Shift estimation between these 2 views of the scene allows controlling the auto-focus (AF) of the main-lens and this method is referred to as Phase-Difference Detection AF (PDAF). Left & right images are then summed to provide a rendered image having twice less pixels than the sensor. This trend is available for smart-phones as well as for DSLR. For instance the Samsung S8 or the Google Pixel 2 have 24 Mega Sub-Pixel sensors producing 12 MegaPixel rendered images, and all Canon DSLRs have a dual-pixel sensor since the EOS 70D. Those sensors define the simplest case of plenoptic sensors popularized by Lytro [1]. Beyond auto-focus they are capable of tight refocusing, aberration correction or bokeh shifting [2]. On top of that, with the growing help of machine learning one can eventually get a depth map using such a sensor and use it to improve bokeh [3].

On the one hand quad-pixel sensors, where each micro-lens covers 2×2 pixels, were studied

by [4] with pure ray-tracing simulations. On the other hand, accurate dual-pixel electromagnetic simulations were carried by [5] and [6]. In this paper we present a method to link the computational imaging world and the quad-pixel's hardware world. It is based on the mix of ray tracing and electromagnetic simulations [7], [8]. We are now able to render more accurate synthetic images with respect to the pixel's response, so we can work on algorithms and anticipate the future prototype. The paper is organised as follows: first we'll show results of quad-pixel electromagnetic simulations, then we'll explain the mixed simulation method itself and describe the validation process. Finally we'll show synthetic images generated with a modified version of PBRT [9, 10], we'll discuss about the sub-pixel size and present applications for the correction of main lens' aberrations. We'll also suggest ideas for solving remaining flaws or improving accuracy.

2. Mixing ray tracing and wave optics

Initially we used 2 softwares independently, Lumerical-FDTD Solution [11] for designing and optimizing the pixel, and PBRT v2 for generating synthetic images and developing algorithms. In a co-design approach, we want to adapt algorithms with respect to the pixel's characteristics and vice-versa. We need a tool capable of generating images and taking the pixel's response into account at the same time. In this section we present a new method for carrying such multi-scale simulations. First, we'll describe the quad-pixel design and related electromagnetic simulations, then we'll explain the methodology and how we link wave optics and ray tracing.

2.1. Finite Difference Time Domain simulations (FDTD)

2.1.1. Quad-Pixel design

We perform FDTD simulations in order to design the quad-pixel stack and optimize some parameters, such as the radius of curvature or the height of the microlens. We are particularly interested in the power absorbed by the silicon of photodiodes, noted P_{abs}^{id} where id is the name of the sub-pixel. The incident light is a linearly polarized plane wave which has a wavelength of $\lambda=550$ nm. Its angle of incidence is decomposed into φ and θ , which are the azimuthal and polar angles respectively ($0 \leq \theta$; $0 \leq \varphi \leq 360$). As we use 2 orthogonal polarizations (0° and 90°) for each simulation setup, P_{abs} is in fact the average of $P_{abs,0^\circ}$ and $P_{abs,90^\circ}$. In Lumerical we use a non uniform grid with an auto mesh refinement setting of 2, which corresponds to 10 Yee cells per wavelength. Boundary conditions on x and y axis are set to Bloch-periodic and we use absorbing PML (stabilized, 256 layers) for the z boundary condition. The general design is based on a back side illuminated (BSI) pixel consisting of a microlens placed on top of planar layers with given thickness: a planarization layer made of transparent resin (500 nm), a green color filter (1 μ m), an passivation layer made of silicon oxide (650 nm) and silicon including the 4 sub-pixels (4 μ m). Deep trench isolation (DTI) inside silicon is used to isolate photodiode from each other [12], and is 100 nm wide. This design and the orientation of φ and θ are depicted in Fig. 1.

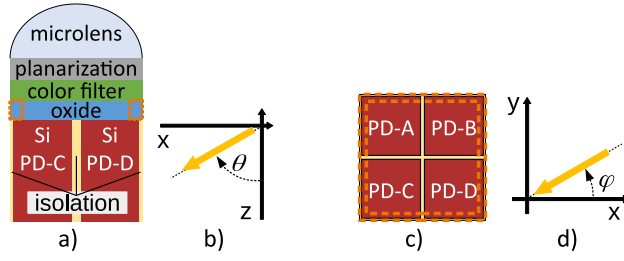


Fig. 1. General design of a quad-pixel sensor. (a) and (c) show the view from the side and from the top (not to scale). Orange rectangles in dashed lines show the tungsten isolation described in section 2.1.3. (b) and (d) show the orientation of the angles θ and φ .

2.1.2. Angular response and performance

First, we simulate a quad-pixel using $1.75\ \mu\text{m}$ sub-pixels, including isolation, so the quad-pixel basis is a square of $3.5 \times 3.5\ \mu\text{m}$. Both the microlens height and radius of curvature (RoC) took 3 values ($2.48\ \mu\text{m}$, $3.00\ \mu\text{m}$ and $3.5\ \mu\text{m}$), leading to 9 different setups: one for each RoC/height combination as shown in Fig. 2(c). With gapless microlenses, $2.48\ \mu\text{m}$ is the minimal RoC and height possible because it is the half-diagonal of the quad-pixel. θ varied from 0° to 40° by steps of 5° and 40° to 60° by steps of 10° , φ is set to 0° . Fig. 2 shows simulation results with a microlens height of $2.48\ \mu\text{m}$ and a radius of curvature of $3.5\ \mu\text{m}$ (Fig. 2(a)) and $2.48\ \mu\text{m}$ (Fig. 2(b)). φ is fixed at 0° so the angle of incidence is in the x-z plane. The sub-pixels C and D are symmetric with A and B thus they are not represented here.

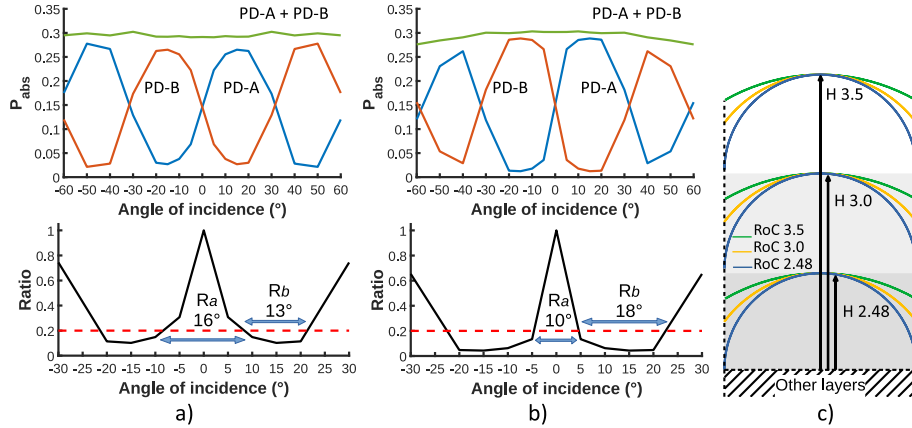


Fig. 2. Angular responses for a microlens height of $2.48\ \mu\text{m}$ and a radius of curvature of $3.5\ \mu\text{m}$ (a) and $2.48\ \mu\text{m}$ (b). Plots at the top show P_{abs}^A , P_{abs}^B and their sum. Plots at the bottom show the rejection ratio between the two sub-pixels for $\theta = -30^\circ$ to 30° . (c) presents the different setups (not to scale) for each combination of RoC and height. The volume between the surface of the microlens and the other layers is filled with the same material.

Following Kobayashi et al. [5], we compute the rejection ratio between signals delivered by the sub-pixels defined as follow :

$$\text{ratio} = \begin{cases} P_{\text{abs}}^B / P_{\text{abs}}^A & \text{if } \varphi = 0 \text{ and } \theta \geq 0 \\ P_{\text{abs}}^A / P_{\text{abs}}^B & \text{if } \varphi = 180 \text{ and } \theta \geq 0 \text{ (equivalent to } \varphi = 0 \text{ and } \theta \leq 0) \end{cases} \quad (1)$$

We use the same threshold of 0.2: the signal is considered good if the sub-pixel of interest receives 5 times the power of the opposite one. Based on this curve (see Fig. 2), we define two ranges: the central part of the curve R_a where the ratio is above the threshold, and R_b where the ratio is below the threshold. R_a is related to "intra-pixel" crosstalk between the 4 sub-pixels, which is higher around the normal incidence. R_b is limited both by "intra-pixel" crosstalk at low angles of incidence and by "inter-pixel" crosstalk, between neighbouring quad-pixels, at high angles of incidence. For instance the simulation in Fig. 2(b) is better than the simulation in Fig. 2(a): in case a) $R_b = 13^\circ$ and $R_a = 16^\circ$, whereas in case b) $R_b = 18^\circ$ and $R_a = 10^\circ$. A high microlens with a long focal length is supposed to be more sensitive with respect to the angle of incidence because the light has more distance to propagate away from the center of the quad-pixel, so it should improve the intra-pixel crosstalk but degrade the inter-pixel crosstalk. The opposite occurs with a small microlens height and a short focal length. In addition to that, a small radius of curvature, which means a more converging lens, should decrease the diameter of the diffraction spot thus improving both intra and inter pixel crosstalk. This is what we tested with the 9 combinations described above.

We optimize the microlens by using the criteria below:

- The acceptance angle (the maximum angle of incidence before crosstalk) must be increased
- The range where the rejection ratio is below a certain level (R_b) must be increased
- The range where the rejection ratio is above a certain level (R_a) must be decreased
- The total power absorbed $P_{\text{abs}}^{\text{tot}} = \sum P_{\text{abs}}$ must be increased.

We found the best design is the smallest microlens height and the smallest radius of curvature.

2.1.3. Decreasing the inter-pixel crosstalk

While the FDTD area is surrounding a single quad-pixel, the boundary conditions are periodic along x and y axis so the crosstalk from the neighbouring pixels is taken into account. For example if we "aim" at the sub-pixel A, i.e. with a positive angle θ , $P_{\text{abs}}^{\text{B}}$ should decrease as θ increases. However this is not the case here: $P_{\text{abs}}^{\text{B}}$ increases again after 20° . This is due to the lack of isolation between each pixel and could be improved with a better light concentration on the pixels. Nevertheless, the maximum absorbed power is reached when the microlens covers the whole surface of the pixel, hence the radius of curvature can not be smaller than the half-diagonal of a pixel (here $2.48\mu\text{m}$ with $1.75\mu\text{m}$ sub-pixels). Thus, the other way to improve performance is to decrease the thickness of the optical stack. In order to investigate on the impact of stack thickness and sub-pixel pitch, we carry out two additional simulations:

1. $1.75\mu\text{m}$ sub-pixel with thinner color filter (750 nm instead of $1\mu\text{m}$ for reference design)
2. $1.4\mu\text{m}$ sub-pixel with a 750 nm color filter and a tungsten isolation between sub-pixels

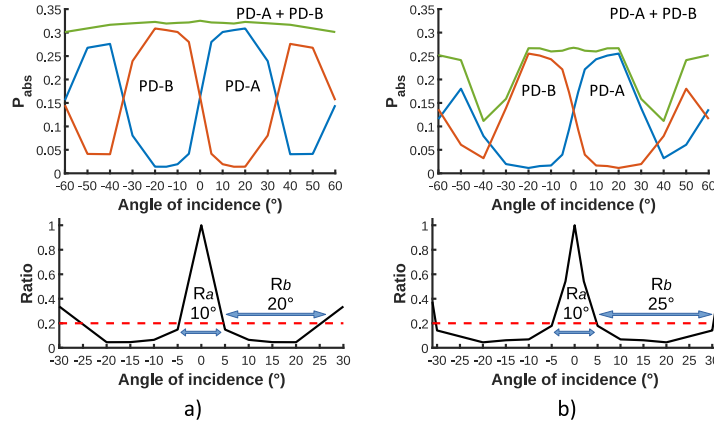


Fig. 3. Angular response of the 3.5 μm pixel with a 750 nm color filter (a) and the 2.8 μm pixel with a 750 nm color filter and inter-pixel tungsten isolation (b).

Fig. 3(a) shows the results for the 3.5 μm pixel and 750 nm CFA and denotes slightly better performance compared to a 1 μm color filter. However, decreasing color filter thickness may degrade color separation. We also simulate a 2.8 μm pixel with tungsten walls immersed in the oxide layer, as shown in Fig. 1. Even if the pixel is smaller, the acceptance angle and the angular response are better than the basic 3.5 μm pixel design. The results are shown in Fig. 3(b). Even if the pixel could be further optimized, for instance with tungsten walls inside the color filter or optical stack thickness tuning, those solutions are constrained by the manufacturing process and this study is beyond the scope of this article.

2.1.4. Chief Ray Angle correction

The acceptance angle determines the maximum aperture of the main lens we can use with the sensor. For example marginal rays from a F/1 lens have an angle of incidence of $\pm 26.6^\circ$, meaning we can optimize the pixel for angles below 26.6° only.

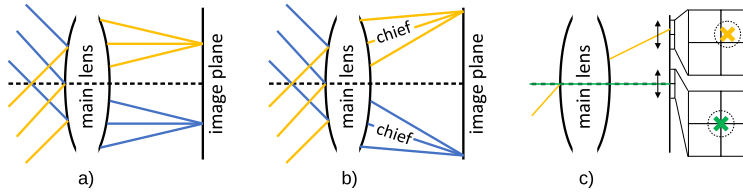


Fig. 4. (a) and (b) compare telecentric and non-telecentric lenses. The cone of light is not centered on the pixel when using a non-telecentric lens and causes vignetting (c).

However this is true only if we consider the pixel on the optical axis or if we use a telecentric main lens (Fig. 4(a)). For non telecentric main lens, the cone of light is centered on the chief ray (Fig. 4(b)). It forms an angle with the image plane which increases as we move away from the optical axis. If the microlens is centered on the pixel, the micro-image is decentered and each sub-pixel does not receive the same amount of light (Fig. 4(c)). In standard sensors, this increases cross-talk and it causes vignetting in the image, but for a plenoptic sensor it is mandatory to re-center the diffraction spot, otherwise the phase signal may be lost. To solve this problem, the usual solution consists in shifting the microlens with respect to the pixel center [13]. The amount of translation depends mainly on the main lens design and some of them, especially smartphone

lenses, are able to restrict the Chief Ray Angle (CRA) below a certain value, typically around 30° .

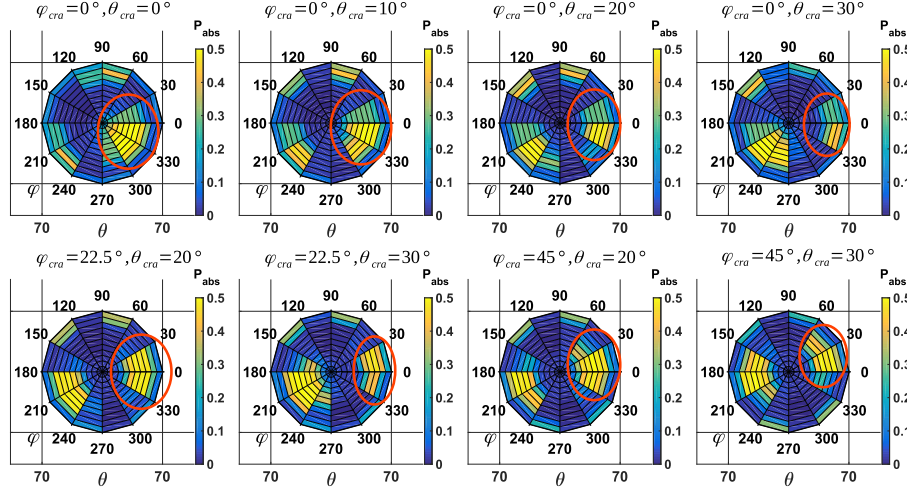


Fig. 5. Angular response of CRA simulation. The first row shows the results with $\varphi_{cra} = 0^\circ$ and the second row shows angular responses for $\varphi_{cra} = 22.5^\circ/45^\circ$ and $\theta_{cra} = 20^\circ/30^\circ$. φ and θ are represented as the angular and radial coordinate respectively.

We perform FDTD simulations using the $3.5\ \mu\text{m}$ pixel and $750\ \text{nm}$ thick color filter. The Chief Ray angle is decomposed into θ_{cra} and φ_{cra} and we optimize the shift for $\theta_{cra} = 0^\circ, 10^\circ, 20^\circ$ and 30° and $\varphi_{cra} = 0^\circ, 22.5^\circ$ and 45° . We complete the data in φ_{cra} using symmetry to finally get data for φ_{cra} to 0° to 360° by steps of 22.5° . θ varies from 0° to 40° by steps of 5° and 40° to 70° by steps of 10° . φ varies from 0° to 360° by steps of 30° . We finally obtain the complete angular responses of each sub-pixel and form a 5D dataset: $(\varphi_{cra}, \theta_{cra}, \varphi, \theta, \text{sub-pixel})$.

Fig. 5 shows P_{abs}^A for some CRA values, especially the cases where θ_{cra} is big. The main lobe is circled in red and the other lobes represent the interpixel crosstalk. The intrapixel crosstalk is not represented here as we only focus on the PD-A and the effects of the microlens' shift. Moreover it is not obvious to find a proper definition of the rejection ratio in two dimensions by taking into account both θ and φ . Considering the photodiode A and the convention of angles described in paragraph 2.1.1, we expect this photodiode to receive flux only if φ is between 270° and 360° and if θ is between 0° and $\arctan(\text{sub-pixel size}/\mu\text{lens focal length})$, neglecting the diffraction for the sake of simplicity. That is what we can see for a centered microlens in Fig. 5. Now if we consider a large angle of incidence ($\theta > 30^\circ$) coming from other directions ($\varphi \neq 0^\circ$) the photodiode A can still receive flux from adjacent quad-pixels. This explains the position of the crosstalk lobes: one at $\varphi = 90^\circ$ from the quad-pixel in the previous row of the sensor, and one at $\varphi = 180^\circ$ from the quad-pixel in the previous column of the sensor. Finally, when the microlens is not centered, all these lobes are shifted and distorted. The data presented here are only the pixel's angular response and are independent of the main lens. The light will not necessarily fall in the crosstalk lobes, it depends only on the main lens CRA and F-number. This can be an issue for wide aperture main lenses and it stresses the importance of interpixel isolation.

172 2.2. Generation of synthetic images

173 2.2.1. Augmented ray tracing

174 Ray tracing is a physically accurate technique for generating synthetic images. It launches rays
175 from all sub-pixels of the camera in random directions and computes the path of light using
176 laws of physics, until it meets a stop condition (light source reached, maximum distance without
177 intersection, absorbing material, bounced to other directions...). Finally, the sub-pixel from which
178 the ray was launched is updated with the color values coded on RGB space. Multiple rays are
179 needed to produce good results without ray-tracing noise, typically more than 16 rays/sub-pixel.
180 This technique simulates many optical effects, like reflection, refraction, diffusion of the scene.
181 We use a custom in-house version of PBRT which also takes into account the main lens and an
182 array of microlenses in front of the sensor. It enables us to simulate a plenoptic sensor and the
183 defects of the main lens such as chromatic aberrations or geometric distortion. The generated
184 images are realistic with respect to the scene and the main lens, but the pixels themselves are
185 still ideal. In this section we propose a method for mixing the ray tracing technique with FDTD
186 simulations described previously. This allows us to generate images which are end-to-end realistic
187 from the scene to the sensor. The main issue is the simulation of the optical diffraction which
188 can not be neglected at pixel scale. A paraxial microlens used in classical ray tracing can only
189 produce a perfect angular response, as depicted in Figs. 6(a) and 6(b). In this geometrical mode
190 the maximum angle of incidence θ_L is equal to:

$$\theta_L = \arctan\left(\frac{\text{sub-pixel size}}{\mu\text{lens focal length}}\right) \quad (2)$$

191 Hence R_a is always 0 and the inter-pixel crosstalk can not be simulated. This is due to geometrical
192 optic laws used in ray-tracing: one ray launched from the left sub-pixel can not go to the left
193 (with a negative angle θ) at the exit of the microlens. However diffraction causes light to fall on
194 both sub-pixels when θ is low, meaning R_a will never be 0 in practice, unless working with much
195 bigger pixels and wider isolation.

196 The sensor will be used for conventional imaging, i.e. under incoherent light, so one sub-pixel
197 will receive the sum of intensities of all rays coming from all directions. At pixel scale we
198 consider that a ray is represented by a plane wave of equal wavelength and equal angle of
199 incidence. Moreover, the phase of electro-magnetic field is already taken into account in FDTD
200 simulations. This means we do not have to take the phase into account, by overloading a ray
201 with a complex number [7], or by making wavefront/rays conversion [8]. In fact, we only have to
202 mimic the diffraction and its impact on absorbed power, we don't need to simulate it directly. All
203 the work is already done via the FDTD simulations during the design process. Ref [14] gives
204 theoretical justifications on the relationship between rays and wave optics using eikonal equation.

205 Our idea consists in splatting the rays on the whole quad-pixel and not only on the sub-pixel of
206 interest, as shown in Fig. 6(c). By doing so, one sub-pixel is virtually able to see rays coming
207 from every direction, regardless of geometrical optic laws. Then we just have to weight the ray
208 with the normalized angular response from the FDTD simulations in Fig. 6(d), with respect to
209 the angle of incidence.

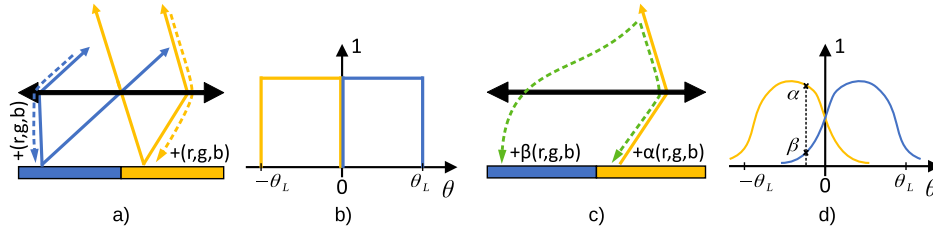


Fig. 6. (a) and (b) show the classic ray tracing procedure and its ideal angular response. (c) and (d) show our modified ray tracing procedure and its angular response.

Here is the algorithm detailed step by step:

- A ray is launched from one sub-pixel to the microlens (right sub-pixel in Fig. 6(c))
- The angle of incidence (φ, θ) is computed just after the microlens. If the CRA is corrected, we also compute ($\varphi_{cra}, \theta_{cra}$)
- The ray propagates through the scene and produces (R,G,B) values
- We read the angular response from FDTD simulations and we compute P_{abs} per sub-pixel for the given angle of incidence, using a 2D linear interpolation (4D if the CRA is corrected)
- (R,G,B) values are added to each sub-pixel weighted by the corresponding P_{abs} (Fig. 6(d))

In PBRT we use ideal microlenses because we must reproduce exactly the given angular response. If we define the microlens with real surfaces then the angular response is convolved with the spot diagram instead of a Dirac function, and that would not be the real angular responses from FDTD. The F-number of one microlens must match the main lens' F-number: it's no use launching rays outside the exit pupil of the main lens, otherwise they are stopped and lost, causing the raytracing noise to increase. In addition to that, the microlenses can be individually shifted in order to correct the CRA. It means when a ray is launched from an off-axis pixel, it naturally goes inside the cone of light centered on the Chief Ray after the microlens.

Performing a 4D interpolation for each sub-pixel and each ray leads to heavy computation. For a quad-pixel sensor, we estimate a performance loss of about 20% for the same amount of rays/sub-pixel compared to classical mode. It is worth noting this method is not restricted to quad-pixel sensors and can be easily generalized for $m \times n$ sub-pixels plenoptic sensors.

2.2.2. Validation of the method

In this section we describe 2 tests which aim at checking the accuracy of our mixed simulation method. We want to reproduce the FDTD angular response with the modified PBRT. We define a sensor made of only one quad-pixel and we launch up to 1048576 rays/sub-pixel. In PBRT, the light source and the camera are transparent to each other, so we are forced to define at least one object with a white lambertian texture in order to reflect the light toward the camera. The light source is a directional light source placed behind the camera at infinity.

In the first experiment we place a 5 mm disk 1 m away from the pixel, supposed to simulate a collimated point source. By moving the disk along x and y axis, we can test the response of the 4 sub-pixels for different angles of incidence. Then we warp the (x,y) data using backward linear interpolation to get a visualisation in polar coordinates, as shown in the previous section.

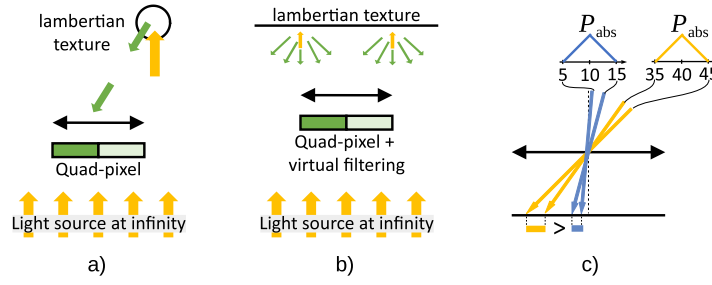


Fig. 7. Description of experiments. Experiment 1 (a) is a moving disk with lambertian texture. Experiment 2 (b) is the infinite plane with lambertian texture and angle filtering. In experiment 2, the illuminated area varies with θ and a normalization step is necessary (c).

In the second experiment we place an infinite plane in front of the camera, so that rays can come from every direction, then we filter the rays by their angle of incidence. It aims at simulating a collimated extended source. Rays are filtered by providing PBRT modified angular response: instead of giving the full real angular response to PBRT, we only put the value corresponding to the angle we want to test. The other values are set to 0. By doing this for all angles of incidence, we can reconstruct the angular response of the pixel. The advantage over the first experiment is that we do not need to warp the data because it is already tested in angle. However we must normalize the data because the illuminated surface is not constant, due to the linear interpolation performed in PBRT. Rays can be launched with an angle θ in $[\theta - \Delta\theta, \theta + \Delta\theta]$ and φ in $[\varphi - \Delta\varphi, \varphi + \Delta\varphi]$. Hence, the illuminated area increases as θ increases and causes the sub-pixel to virtually receive more light than expected. This problem is depicted in Fig. 7(c).

The normalization factor is the area of the illuminated surface, given by:

$$N = \underbrace{2\Delta\varphi}_{(a)} \times \underbrace{f_{\mu\text{Lens}}^2 [\tan^2(\theta + \Delta\theta) - \tan^2(\theta - \Delta\theta)]}_{(b)} \quad (3)$$

where (a) is the fraction of area delimited by $\varphi \pm \Delta\varphi$ and (b) is the fraction of disk delimited by $\theta \pm \Delta\theta$.

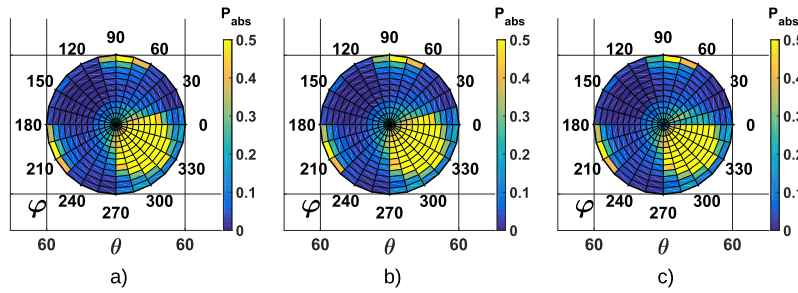


Fig. 8. Validation of our method showing normalized angular response of sub-pixel A in different cases: (a) FDTD angular response, (b) Experiment 1 (moving small disk), (c) Experiment 2 (fixed large texture)

Finally we can compare the angular responses from the FDTD simulations and the 2 validation experiments. Fig. 8 shows the angular response of sub-pixel A with the $2.8\mu\text{m}$ quad-pixel for the FDTD simulation (a) and the 2 experiments (b) and (c). The results are computed for the same

258 angles and normalized to 1. The angular response from experiments 1 and 2 are very similar
 259 to the FDTD ones, even the cross talk is simulated. There are some local differences due to
 260 the bilinear interpolation or the random process involved with ray-racing (plus eventually the
 261 warping in experiment 1). Overall we consider the method is good enough for our future work
 262 but there are some aspects which still could be improved. They are discussed in next section and
 263 in conclusion.

264 3. Applications to quad-pixel sensor

265 3.1. Example of synthetic images

266 In this section we present some images generated with our method implemented in PBRT-v2. A
 267 5.6×5.6 mm sensor is simulated using $1.75 \mu\text{m}$ sub-pixels for a total of 1600×1600 microlenses.
 268 We use the lenses described in patent US 8,320,061 B2 (wide angle) and US 9,316,810 B2
 269 (periscope-like), respectively called *ML-1* and *ML-2* for convenience. They are represented in
 270 Fig. 9, *ML-2* is unfolded but the mirror is still drawn. The chief ray angle is corrected for the 2
 271 lenses according to the optical software's specifications. We launch up to 128 rays/sub-pixels. In
 272 this section and the following ones, the synthetic images are not demosaiced: PBRT already gives
 273 (R,G,B) value for each sub-pixel. We just want to illustrate the simulation method independently
 274 of traditional image processing. We use the ISO-12233 chart and a custom chart made of tiled
 275 alphabet texture shown in Fig. 9(c). It is mainly used for qualitative sharpness estimation as the
 276 letters are a bit less artificial than straight lines, but it is also useful to study chromatic aberrations.
 277 The San-Miguel environment is also used to render real scene instead of test charts (available
 278 here: <https://www.pbrt.org/scenes-v2.html>).

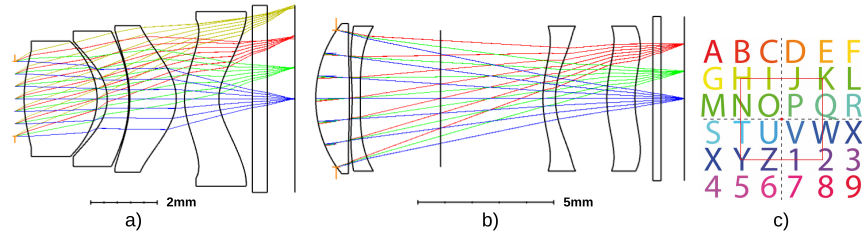


Fig. 9. (a) *ML-1* from US8320061 B2, (b) *ML-2* from US9316810 B2, (c) abcd texture.

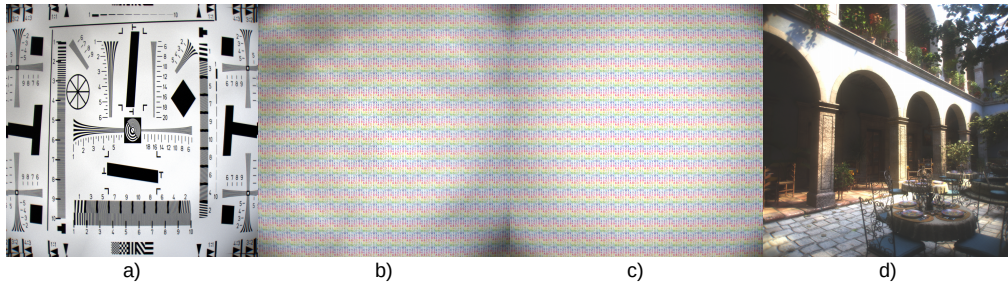


Fig. 10. Examples of synthetic images using a quad-pixel sensor made of $1.75 \mu\text{m}$ sub-pixels and *ML-1* as main-lens. (a) and (b) are the SAI-A of the ISO-12233 and "abcd" test chart. (c) and (d) are the sum of the 4 SAIs of the abcd test chart and San-Miguel scene respectively.

279 When working with a plenoptic sensor we usually define the sub-aperture images (SAI), formed
 280 with every sub-pixel having the same position under the microlens. For instance the SAI-A is

made of every sub-pixel A and the final image is the sum of the 4 SAIs [4]. Figs. 10(a) and 10(b) are the SAI-A of two test charts at 750mm, Figs. 10(c) and 10(d) are the sum of the four SAIs of the abcd test chart and the San-Miguel scene.

We notice variations in intensity on the SAIs where bright lines correspond to the CRA grid points in FDTD simulations. Our method requires many electromagnetic simulations to be accurate, especially because of the 2D/4D linear interpolation performed in PBRT. Usually for a given and fixed CRA, the bilinear interpolation is good enough with $\Delta\theta = 5^\circ$ and $\Delta\varphi = 30^\circ$, leading to 132 FDTD simulations. Then we can render a small portion of the sensor where we consider the CRA is almost constant. However as soon as we want to simulate the whole sensor the CRA must vary continuously and we must perform a 4D interpolation on the angular response. For example in the first section one can notice that $\Delta\varphi_{cra} < \Delta\varphi$ (22.5° and 30° respectively), meaning the CRA sampling does not respect Shannon criteria. This lead to interpolation artefacts so we have to refine the grid. The main issue is that one FDTD simulation takes 20min using $0.8\mu\text{m}$ sub-pixels ($121 \times 121 \times 848 \approx 12M$ gridpoints) and up to 1 hour for $1.75\mu\text{m}$ sub-pixels ($262 \times 262 \times 872 \approx 60M$ gridpoints). Those simulations used to run with a number of core between 8 and 20, a frequency between 2.8GHz and 3.2GHz, and an amount of RAM between 12 and 32 Go, depending on the availability of the clusters. If we use the following grid and $0.8\mu\text{m}$ sub-pixels: $\theta_{cra} = 0:10:30 \mid \varphi_{cra} = 0:22.5:45 \mid \theta = 0:5:50 \mid \varphi = 0:11.25:348.75$ then we will have to perform 4224 simulations, in other words almost 2 months of computation. To reduce the number of simulations, some solutions may be possible, such as: pre-interpolate data outside PBRT, find a 4D function basis and fit angular response on it, or define a meta-model. The final image however has an almost constant intensity across the field of view because the sum of P_{abs} is constant and preserved through linear interpolation. This problem is not to be confused with the natural vignetting of the main lens in the corners visible on the final image. We may add additional FDTD simulations in the future to get cleaner SAIs.

Speaking of performance, one PBRT render of a real scene using a 3400x3400 sub-pixels sensor (1700x1700 microlenses) with 32 rays/sub-pixels, in diffractive mode, with the ML-1, takes about 12 hours to run. We use a 6 cores/12 threads processor at 3.5GHz with 16Go of RAM. PBRT scales very well with the number of cores and it would be easy to compute multiple renders during a night with a more powerful computer. The method described above is useful in the case of small pixels because the effects of diffraction are not negligible. Fortunately, FDTD simulations scales with the volume of the mesh so they are faster with small pixels and it reduces the overall computation cost of the method. One could also use GPU acceleration for both FDTD and raytracing simulations and decrease the runtime of the proposed method. However neither Lumerical nor PBRT offer GPU acceleration and we can not make performance comparison.

3.2. Discussion: impact of the sub-pixel size and wavelength dependency

3.2.1. Sub-pixel size

In order not to lose too much resolution by using quad-pixels, one would like to decrease the sub-pixel size as much as possible. Beside the study above, we have also studied the impact of the sub-pixel size on the performance. The SAIs produced by an ideal quad-pixel sensor, where the range R_a is zero, are the images formed by exactly one quarter of the main lens' pupil. However the intra-pixel crosstalk causes each SAI to receive light from the 3 other pupil's quarters and therefore the disparities are harder to compute with accuracy. The worst case happens when the sub-pixel receives the same flux almost independently of θ : SAIs are identical and the phase signal is lost. We have simulated various sizes of quad-pixels from $1.2\mu\text{m}$ to $6\mu\text{m}$. The different designs were not finely optimized but it gives a general trend. In particular, the range R_b for $4\mu\text{m}$ and $6\mu\text{m}$ pixel pitch could have been better. The $3.5\mu\text{m}$ optimized pixel is shown using red stars. All the results are grouped in Fig. 11 and shown in terms of angular ranges R_a and R_b .

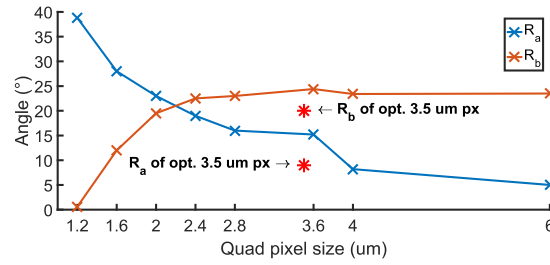


Fig. 11. Impact of sub-pixel size on angular response.

Let's remember that the phase signal is extracted more easily if R_a is small (intra-pixel crosstalk) and R_b gives the maximum angle of incidence possible (inter-pixel crosstalk). We are now able to study the impact of the intra-pixel crosstalk on the final images. We use the *ML-1* focusing at 750 mm from the camera. The scene is the "abcd" chart placed at 500 mm. It is slightly out of focus in order to see shifts between each SAIs, but keeping the letters quite sharp and not completely blurred. We use 3.5 μm and 1.6 μm quad-pixels in both classical and diffractive mode.

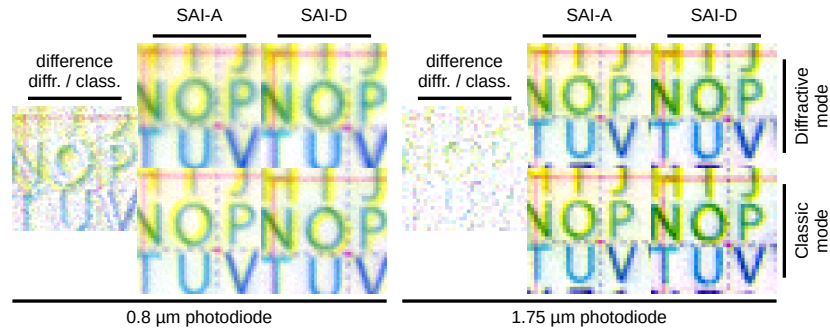


Fig. 12. Sub aperture image of sub-pixel A and D at the center using 0.8 μm and 1.75 μm sub-pixels, in classical and diffractive mode. The difference between the classical and the diffractive mode of SAI-A is shown on the left.

Those different cases are illustrated in Fig. 12. With the 1.6 μm quad-pixel shifts are still visible, confirming that a stereo signal can be extracted. However the blur caused by the intra-pixel cross talk becomes visible and it could be a limiting factor for precise auto-focus or accurate depth estimation. On the contrary, the 3.5 μm quad-pixel is less impacted. The best compromise between sensor resolution and phase-signal extraction seems to be sub-pixels between 1.0 μm and 1.4 μm . A 1/1.7" quad-pixel sensor with 1 μm sub-pixels will have an 11 MPix final image. That is less than common smartphone sensors but it offers light field abilities like refocusing, aberration correction, bokeh improvement or passive depth estimation.

3.2.2. Wavelength dependency

In this section we present the variations of the angular responses with respect to the wavelength. We work with 0.8 μm sub-pixels at 400, 532 and 700 nanometers with blue, green and red color filters. We test the wavelength dependency for a centered microlens and a shifted microlens for a CRA of $\varphi_{cra} = 0^\circ$, $\theta_{cra} = 25^\circ$. The results are presented in Fig. 13. As expected, the angular response is worse at 700nm than at 532nm because the light is more diffracted. Compared to the wavelength, the microlens is only twice bigger and one subpixel has almost the same size. On

the other hand, the angular response is at 400nm better than at 532nm for the same reasons: the microlens and the sub-pixel are bigger compared to the wavelength.

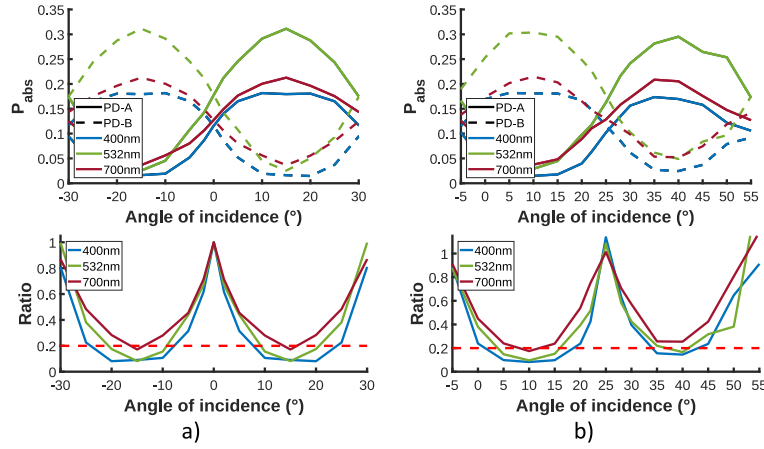


Fig. 13. Wavelength dependency with a $0.8\mu m$ sub-pixels for a centered microlens (a) and a shifted microlens for CRA correction of $\varphi_{cra} = 0^\circ$, $\theta_{cra} = 25^\circ$ (b).

As said in section 2.1.4 when the CRA is corrected the angular response is shifted and distorted. This is particularly visible at 700nm on the ratio plot, where the right part of the curve exceeds the threshold of 0.2. Depending on the usage of the sensor (autofocus only, depth estimation, aberration correction...), it could be an issue or not. We want to stress the fact that the threshold of 0.2 is arbitrary. Having a ratio above this threshold does not mean that the algorithms will stop to work immediately, they will only become more or less accurate. The wavelength dependency is not taken into account in PBRT, we only use the 550nm angular response. It's mainly because we have not carried a full set of FDTD simulations, with CRA correction, at multiple wavelengths.

3.3. Main-lens Aberration corrections

In this section we use an algorithm which aims at correcting the main lens aberrations. If the object is out of focus, its image does not have the same position on each SAI. This translation is the disparity ρ and the auto-focus module of a camera is based on this phase signal.

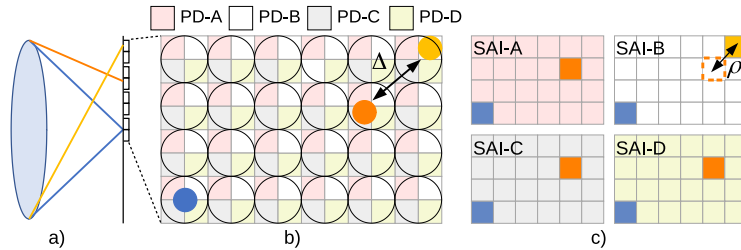


Fig. 14. Main lens' aberrations seen by a quad-pixel sensor. (a) illustrates one aberration of the main lens at one corner (orange and yellow rays). Blue rays are non-aberrant rays at the center. (b) shows the rays falling on different quad-pixels (colored disks). (c) shows the 4 SAIs and the impact of the rays on the sub-pixels. The position difference Δ in sensor space translates into the disparity ρ in image space.

However even if the object is in focus, local shifts can still appear due to the main lens' aberrations (Fig. 14). We can compute these local disparities and then correct the aberrations [3],

hence improving the image quality. The improvement depends on the main lens' design as demonstrated by [15]. Generally speaking, increasing the number of sub-pixels enables to correct more complex aberrations. It is worth noting that the correction is useless when the spot diagram is smaller than one microlens. This case will become rare, as submicron pixels become more and more common in smartphones. In order to correct aberrations, we take an image of a heavily textured surface as noise pattern for instance, which is perfectly in focus. Then we use a block-matching algorithm to compute disparities between a SAI and the reference, which can be one channel of one of the SAIs, or one channel of their sum. We obtain a disparity map made of ΔX and ΔY per sub-pixel and per channel. Finally we warp each SAI using Lanczos interpolation functions according to the disparity map and we get better SAIs and a better image quality. The advantage of this method is it can work without knowing the main lens' design. This is also needed for accurate depth map computation as we need to remove the local shifts which are not related to depth information. We apply the algorithm on generated images using $0.8\mu m$ sub-pixels and we compare the results between the classic mode and the "diffractive" mode of PBRT. Three cases are simulated: *ML-1* in classic mode, *ML-1* in diffractive mode, and the *ML-2* in diffractive mode with manufacturing defects (Table 1).

Element	Misalignment
1	$\Delta z = 1.5\mu m$
2	$\Delta x = 6\mu m, \Delta y = 6\mu m$
mirror	tip = 0.625° , tilt = 0.25°
3	$\Delta x = 9\mu m, \Delta y = 5\mu m$, tip = 0.2° , tilt = 0.2°
4	$\Delta x = 4\mu m, \Delta y = 2.5\mu m$

Table 1. Manufacturing defects for experience 3 with *ML-2*

A flat texture is placed at 750 mm. The angular response of pixels and the CRA correction are both taken into account. We only simulate a small area on the sensor of 150×150 microlenses at the center and for $\theta_{cra} = 10^\circ / \varphi_{cra} = 45^\circ$ (called position 2). By doing so we gain computation time and that overcomes the CRA 4D interpolation issue, considered the CRA is constant on such a small portion of the sensor. The results are grouped in Fig. 15.

The *ML-1* is very good at the center so the gain is minimal either in classic or diffractive mode. However a difference occurs between the two modes at position 2. In classic mode we can see visible improvements (circled in red) whereas in diffractive mode, the gain still exists but it is not as impressive (circled in blue). Gains in resolution are given in Table 2, using the sharpness metrics proposed by [16] and [17], plus the average of the gradient in both directions. The geometrical mode produces a perfect angular response that makes each SAI to see only one quarter of the main lens spot diagram. Disparities appear locally and are easy to compute. The algorithm gathers the corresponding pixels at the same position and the sharpness is improved. In diffractive mode the SAIs see a common flux which comes from the center of the main lens, corresponding to the range R_a , and the phase signal only comes from the edges of the main lens, corresponding to the range R_b . If the common flux is predominant each SAI sees the whole spot diagram and the same aberrations, meaning there is no disparity to compute at all. This is not totally the case here, but that explains why we barely see improvements with $0.8\mu m$ sub-pixels and *ML-1*. It also depends on the nature of the aberrations, some of them are easier to correct than others. For exemple the aberrations caused by the manufacturing defects of *ML-2* are better corrected (circled in green) than the "natural" aberrations of *ML-1*.

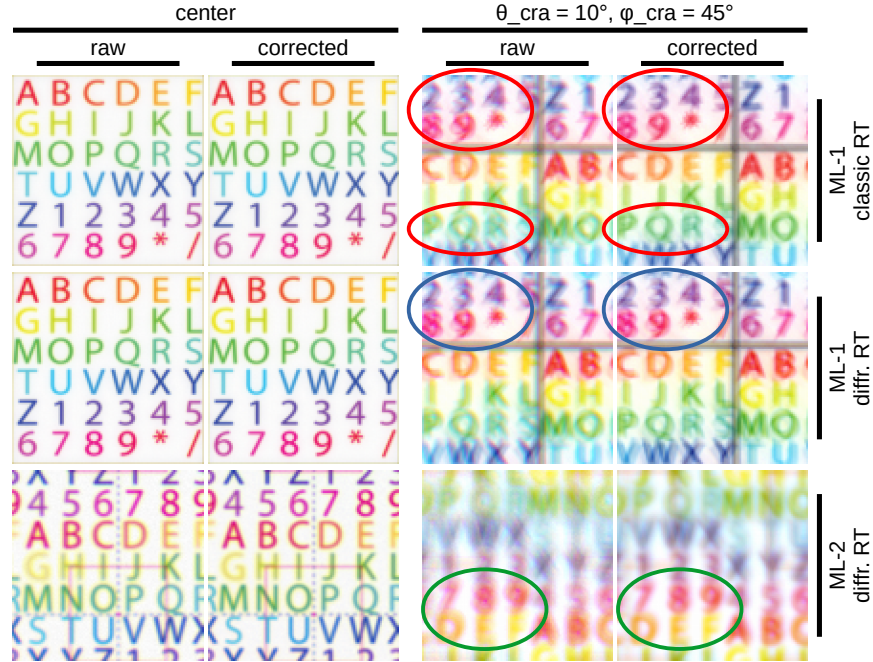


Fig. 15. Results of the aberration correction algorithm for different setups, at the center and for $\theta_{cra} = 10^\circ / \varphi_{cra} = 45^\circ$. The three setups are: *ML-1* in classic mode, *ML-1* in "diffractive" mode and *ML-2* in diffractive mode with simulated manufacturing defects.

		Center			$\theta_{cra} = 10^\circ / \varphi_{cra} = 45^\circ$		
	metric	raw	corr.	gain	raw	corr.	gain
<i>ML-1</i> class.	gradient	0.5048	0.5079	+0.6%	0.2573	0.30	+16.2%
	De [16]	0.2378	0.2384	+0.25%	0.0703	0.0875	+25.5%
	Wee [17](E^{-12})	0.5700	0.5811	+1.96%	0.3415	0.4027	+17.9%
<i>ML-1</i> diff.	gradient	0.5428	0.5424	-0.09%	0.3361	0.3672	+9.3%
	De	0.2534	0.2514	-0.79%	0.0909	0.1055	+16.1%
	Wee (E^{-12})	0.7657	0.7696	+0.5%	1.132	1.210	+6.94%
<i>ML-2</i> diff.	gradient	0.5530	0.5823	+5.3%	0.2173	0.2398	+10.3%
	De	0.2059	0.2283	+10.9%	0.0361	0.0465	+28.8%
	Wee (E^{-12})	1.018	1.132	+11.1%	0.4398	0.5571	+26.7%

Table 2. Summary of image quality gains

4. Conclusion

In this paper we described a new method for generating synthetic images by mixing wave optics simulation and ray tracing simulation. We focused on quad-pixel sensors but the method can be easily generalized for $m \times n$ sub-pixels plenoptic sensors. After designing the pixel with the FDTD software, whose results are presented in the first part, we can get the angular response of each sub-pixel. These data are put in the ray tracing software and this enables us to simulate the diffraction effects on the generated images. The method has been described and validated in the second part. However some flaws remain and the method can still be improved further. First,

we mainly use green light ($\lambda = 550$ nm) in FDTD simulations but the angular response may vary with the wavelength. It is not taken into account yet in PBRT and our ray-tracing simulations may not be totally accurate because all the rays share the same angular response independently of their wavelength. This leads to the second problem: we only use green color filter in FDTD simulations due to the chosen wavelength and the periodic boundary conditions. This means the interpixel crosstalk is probably higher than expected. For example if we consider a sensor with a Bayer pattern, the green crosstalk should be filtered by the adjacent color filters, which are red and blue. In our case, we are in fact simulating the worst scenario. Finally, the interpolation issue explained in section 3.1 is yet to be solved.

The last section illustrates the method by showing examples of synthetic images and presents two applications: a small study on the impact of the sub-pixel size and the correction of the main lens aberrations. This tool simulates the scene, the main lens and the sensor, meaning we are able to co-design the main lens, the pixel architecture and the image processing algorithms with respect to each other. A quad-pixel sensor offers new possibilities in terms of computational imaging, but as shown in the last section the co-design approach is really important to get the best imaging pipeline, especially with small sub-pixels. We are also going to use this tool to generate image database that will be our ground truth in terms of depth map and demosaicing. Then we will be able to train machine/deep learning networks and eventually get even better results than traditional algorithms.

References

1. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Tech. rep., Stanford University (2005).
2. R. Winston, "Canon u.s.a., inc., whats new eos 5d mark iv dual pixel raw images," <https://www.usa.canon.com/internet/portal/us/home/learn/education/topics/article/2018/June/Whats-New-EOS-5D-Mark-IV-Dual-Pixel-RAW-Images> (2016).
3. N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy, "Synthetic depth-of-field with a single-camera mobile phone," *ACM Trans. Graph.* **37**, 64:1–64:13 (2018).
4. B. Vandame, V. Drazic, M. Hog, and N. Sabater, "Plenoptic sensor: Application to extend field-of-view," in *2018 26th European Signal Processing Conference (EUSIPCO)*, (2018), pp. 2205–2209.
5. M. Kobayashi, M. Ohmura, H. Takahashi, T. Shirai, K. Sakurai, T. Ichikawa, H. Yuzurihara, and S. Inoue, "High-definition and high-sensitivity cmos image sensor with all-pixel image plane phase-difference detection autofocus," *Jpn. J. Appl. Phys.* **57**, 1002B5 (2018).
6. S. Choi, K. Lee, J. Yun, S. Choi, S. Lee, J. Park, E. S. Shim, J. Pyo, B. Kim, M. Jung, Y. Lee, K. Son, S. Jung, T. Wang, Y. Choi, D. Min, J. Im, C. Moon, D. Lee, and D. Chang, "An all pixel pdaf cmos image sensor with 0.64 μ m $\times$ 1.28 μ m photodiode separated by self-aligned in-pixel deep trench isolation for high af performance," in *2017 Symposium on VLSI Technology*, (2017), pp. T104–T105.
7. D. V. Johnston and P. N. G. Tarjan, "Cs348b final project: Ray-tracing interference and diffraction," Tech. rep., Stanford University (2006).
8. N. Lindlein, "Simulation of micro-optical systems including microlens arrays," *J. Opt. A: Pure Appl. Opt.* **4**, S1–S9 (2002).
9. "Pbrt v2," <https://www.pbrt.org/>.
10. M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation* (Morgan Kaufmann Publishers Inc., 2016), III ed.
11. "Lumerical ftdt solution," <https://www.lumerical.com/products/ftdt-solutions/>.
12. T. Arnaud, F. Leverd, L. Favennec, C. Perrot, L. Pinzelli, M. Gatefait, N. Cherault, D. Jeanjean, J.-P. Carrere, F. Hirigoyen, L. Grant, and F. Roy, "Pixel-to-pixel isolation by deep trench technology: Application to cmos image sensor," in *IISW*, (2011).
13. G. Agranov, V. Berezin, and R. H. Tsai, "Crosstalk and microlens study in a color cmos image sensor," *IEEE Transactions on Electron Devices* **50**, 4–11 (2003).
14. M. Born and E. Wolf, *Principles of Optics* (Cambridge University, 1999), VII ed.
15. R. Ng and P. M. Hanrahan, "Digital correction of lens aberrations in light field photography," in *International Optical Design*, (Optical Society of America, 2006), p. WB2.
16. K. De and V. Masilamani, "Image sharpness measure for blurred images in frequency domain," *Procedia Eng.* **64**, 149 – 158 (2013). *International Conference on Design and Manufacturing (IConDM2013)*.
17. C.-Y. Wee and R. Paramesran, "Image sharpness measure using eigenvalues," in *2008 9th International Conference on Signal Processing*, (2008), pp. 840–843.