



Conception d'un capteur CMOS intégré pour la capture d'image plénoptique et traitement d'images associé

Guillaume Chataignier

► To cite this version:

Guillaume Chataignier. Conception d'un capteur CMOS intégré pour la capture d'image plénoptique et traitement d'images associé. Optique [physics.optics]. Université Grenoble Alpes [2020-..], 2021. Français. NNT : 2021GRALY071 . tel-03597692

HAL Id: tel-03597692

<https://theses.hal.science/tel-03597692>

Submitted on 4 Mar 2022

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.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE GRENOBLE ALPES

Spécialité : Physique de la Matière Condensée et du Rayonnement

Arrêté ministériel : 25 mai 2016

Présentée par

Guillaume CHATAIGNIER

Thèse dirigée par **Jérôme VAILLANT**, Directeur de
Recherche, CEA-Leti, et
codirigée par **Benoit VANDAME**, ingénieur-chercheur,
Interdigital R&D France

préparée au sein du **Laboratoire d'Electronique et de
Technologie de l'Information (CEA-Leti)**
dans l'**École Doctorale de Physique de Grenoble**

**Conception d'un capteur CMOS intégré
pour la capture d'image plénoptique et
traitement d'images associé**

**Development of an integrated CMOS
sensor for plenoptic image capture and
associated image processing**

Thèse soutenue publiquement le **18 novembre 2021**,
devant le jury composé de :

M. Andrès ALMANSA

Directeur de Recherche, CNRS Ile de France Villejuif, Rapporteur

M. Guillaume DRUART

Ingénieur HDR, ONERA, Rapporteur

Mme. Christine GUILLEMOT

Directrice de Recherche, CNRS Bretagne et Pays de la Loire,
Examinatrice

Mme. Neus SABATER

Ingénieure HDR, Interdigital R&D France, Examinatrice

M. Eric LACOT

Professeur des Universités, Université Grenoble Alpes, Président du jury
et Examineur



Silicon may be doped N or P, PhD student is doped coffee

Résumé

Un capteur plénoptique permet de capturer un champ de lumière c'est-à-dire d'échantillonner spatialement et angulairement les rayons lumineux provenant d'une scène, contrairement à un capteur d'image classique qui permet uniquement un échantillonnage spatial. Cela offre des possibilités plus poussées en termes de traitement d'images. On peut par exemple refaire la mise au point après la prise de vue, corriger les aberrations de l'objectif, ou calculer une carte de profondeur de manière passive. L'échantillonnage angulaire est produit en plaçant plusieurs sous-pixels sous une même microlentille du capteur. Cette thèse concerne un capteur quad-pixel où une microlentille recouvre un patch de 2×2 sous-pixels. Il s'agit de l'évolution des capteurs "dual-pixels" développés par Canon en 2012, principalement utilisés pour contrôler l'autofocus de leurs appareils photographiques. Elle a pour objectif de tisser des liens entre deux des trois aspects du système d'imagerie, les pixels d'une part, et le traitement de l'images d'autre part. L'optique utilisée pour produire l'image sur le capteur est laissée de côté. Je présente d'abord des simulations au niveau des pixels, principalement en utilisant des sous-pixels de 1.75 microns. A partir du critère de performance défini, je présente des pistes d'amélioration des pixels. En me basant sur ces simulations, j'ai modifié un outil de rendu d'images en lancé de rayons qui permet de prendre en compte la diffraction dans les microlentilles. Je montre que la diffraction due à la microlentille dégrade les performances des algorithmes, je prends pour exemple le cas de la correction des aberrations de l'objectif principal. Cet outil a de plus permis de générer des images puis d'étudier les algorithmes de traitement d'images. Je présente plusieurs pistes d'utilisation d'un quad-pixel à l'aide des techniques d'apprentissage profond et de réseaux de neurones, comme par exemple le dématricage couleur ou la correction de la diffraction. Enfin, je présente et caractérise un prototype quad-pixel réalisé dans le cadre de cette thèse.

Abstract

A plenoptic sensor enables the capture of a light field : it spatially and angularly samples the light rays coming from a scene, unlike a conventional image sensor which only allows spatial sampling. This offers more possibilities in terms of image processing. For example, one can refocus the image after its acquisition. It is also possible to correct lens' aberrations, or calculate a depth map in a passive way. The angular sampling is achieved by placing multiple sub-pixels under a single microlens of the sensor. This thesis focuses on the case where a microlens covers a patch of 2×2 sub-pixels, called a quad-pixel. This is the evolution of the so-called dual-pixel developed by Canon in 2012, mainly used to drive the autofocus of their cameras. Its object is to link two of the three aspects of the imaging system, the pixels on the one hand, and the image processing on the other hand. The lens used to produce the image on the sensor is not studied here. First, I present simulations at pixel level, mainly using 1.75 microns subpixels. Given the defined metric performance, I present some solutions in order to improve the performance of the pixels. Based on these simulations, I decided to modify a ray tracing renderer in order to take into account the diffraction in microlenses. It then made it possible to generate synthetic images and thus study the image processing algorithms. I show that microlens diffraction tends to degrades the algorithms' performance by studying the case of aberration correction of the main lens. I also present multiple ways to use a quad-pixel, using deep learning techniques and neural networks, such as debayering or diffraction removal. Finally, I present and characterize a quad-pixel prototype made especially for this thesis.

Remerciements

Je tiens en premier lieu à remercier mes deux formidables directeurs de thèse, l'un au CEA et l'autre à Interdigital. Un grand merci pour votre bienveillance, votre soutien indéfectible et vos précieux conseils durant ces trois (presque quatre ?) années ! J'ai beaucoup appris à vos côtés, je vous en suis infiniment reconnaissant. Je remercie également les deux équipes dans lesquelles j'ai travaillé, celle du CEA et celle d'Interdigital, qui m'ont accueilli, nourri, logé¹, avec qui j'ai eu des discussions passionnantes, et surtout qui ont supporté, tour à tour selon mes déplacements, mes blagues vaseuses. En particulier, merci pour vos conseils en programmation, en traitement du light-field, en Lumerical, et merci pour l'aide sur la caractérisation du prototype. Merci aussi aux *Zesteurs* pour les invitations au Zest (ou ailleurs, mais quand même principalement au Zest, on ne va pas se mentir) et qui m'ont très certainement aidé à ne pas devenir timbré pendant les périodes de confinement/déconfinement successives. Je remercie également les membres du jury d'avoir accepté d'évaluer mon travail de thèse. Merci aux deux rapporteurs d'avoir accepté de lire le présent manuscrit, et merci aux examinateurs d'avoir accepté d'assister à ma soutenance à Grenoble. Les différentes discussions étaient très intéressantes et enrichissantes. Je m'excuse pour l'absence d'un véritable "pot-de-thèse", la soutenance ayant eu lieu dans un contexte particulier commençant par C et finissant par ovid. Je remercie aussi les membres de mon Comité de Suivi Individuel pour leur suivi année après année, et leurs conseils sur les lignes directrices, le planning, et la bonne tenue de mon projet. Je tiens enfin, évidemment, à remercier ma famille et mes amis qui m'ont soutenu tout au long de ma thèse : merci pour leur relecture, les sorties photos improvisées, les voyages organisés à l'arrache (en particulier un où l'on dormait 2h par nuit, une expérience enrichissante s'il en est), les sessions jeux vidéos, les sorties bateau, plongée, plage, restau, bar... Merci pour tout !

D'aucuns auraient remarqué² que je n'ai pas donné de prénoms ni de noms, et que ces remerciements sont relativement impersonnels. C'est là que la blague arrive : j'ai voulu rendre ça plus ludique. Tout se *joue* à la page suivante. Et puis comme j'ai une mauvaise mémoire et que j'ai probablement oublié des gens, je pourrais répondre : "C'est juste que tu ne t'es pas trouvé".

Bon³ courage⁴ !

1. Avec plus ou moins de succès, je suis ainsi devenu le SBF (Sans Bureau Fixe) du LIS

2. Ceci est sans doute la tournure de phrase la plus littéraire de ce manuscrit, profitez en

3. Oui j'ai la solution, non vous ne l'aurez pas.

4. Bravo aux personnes qui trouveront toutes les références cachées

VJJYEAXPQQZKVADGALMGROXYOSSICDXSEPLDOLUCIE
CDJCRBBLUEGEIEAEBINRNNBZAVKMIRQYXVTITOUANB
KGVFNVTZENGMOXTPEAHUGL THERAPIETAXIYLSTYCFL
TSAYHWQONOPPVRLMHKEZVACVBUIAVARMVALTERAHGI
CEGHSONLTVINCENTDLIZSUJSQCJNENOEYTJULIARVN
FBMUXIRQIVOCYZEAOHRUORFKJEROMEYZFALSJOIIQD
PAOKHWYANJKQPXRCXVSNRECQNGBEBKPZINSGBTYSUG
ZSEHUCLEMENTUIIXZAFOMNOMAHNNZLCAHGERTHBTCU
UTNPDXIDYMXMDHCBFNZSPTNORBERTADNULCEBLAIHA
KIXMFDSANDRAAPXAADFEUYJAUDOJGNEINERESDJNAR
WEOATLGKAMEGVQQOTADEXDMHAGDFRTMNLDMNRMIEBD
VNEXJOSNTAAOIUYEMMXVEGEKUJVHTOOEIRUDRGBLEI
XGMEOPBYUPTYDPOUAEGVXNXHVIDHQINKROJAVRVDF
NUONSYMGEPDOYIOAISABATEREBDTHNSNOUZYTSGRN
DIUCQKZPODPNOEIFEKISKZGDSGLRMEAXNADAHQTUAH
ILIENTXRXANJDTRKWXQQZFJVLTZZIDSNCMSHISHIAWV
DLLOMWMYSFXCJREEDUOYXLOHAJYSDDDNALVVROWRGL
IELEQUAQZJCPZEQYNWSFDWSBGMFTNHWXIEHAXGPTUS
EMEWVCHORALIEDALILBXICDNIVWAYBIHDCWLFXMMSB
ROTHESYXASBLPNWOPJPBBWOMAZWNZCZLEXPEODYVAF
ITVJRNXXKHOGZLQVZHMXXCOQSFICVBKTATNEPRUSNVRF
NKYECOVMFAXAQJSWULHHUPWUWRLCBNTRFDPAIXKTLNU
IINAILAMIXGMAOFLARVIPYMXETVZGMDEJILEOIHSAJ
CERNNNEUSUANRFGMVGITAPKGSVPVJKSTUCLJKLOGUU
OMLNECNEHVSANLDEUUDHPZNTKOLIVIERSAIMSLSODD
MMXGDOBXEZPPOSAPLIWEADBTIVFMTASQTCEHAEMBNA
AAHSTVESRXACJHALXLDCBOCAEXTVTJEXIKLYNTEORS
TMHSVANPOBRGNIKKELQVZQABXEWPNMHRNMFOAGCQAP
MEXHLIODGCDAAAYHEEAAMHETTHIBAUTFIENOFEXFKXR
ALRQGLIGSXWMHAHYBUCHTPHDVJJFEPVCYPXPEJSEEI
TIVOSLTFJYIAGTAREMHJLIEWMILREMYWRRHZXZTHLE
AEBOVANESSAMSJLIFEGYRDRKVELERTAKPOWERWOLFS
HYOEYNVOJYHAECEDRICCERISGQEZOLTANFQHHGJKFT
WHFOHTHWVKINSOSCAGFJBDNAHEULBEUKALMANSAGLF
WOB JPMZZSKKYKYTYNMIRMIEMARIONFUZDYBDVKECUK
ZXYARSXWBOBINFOCÇTHESARDSDXAKYISANDRINEPQC
SHAKAPONKZBLAQRPOXAVIERDANGMELANIELACOTPJS
WNANDRESYEIFJFMPIERREYVESCWVWVURNVCVPZMRFRX
MATTHIEUYIWTIIVRSIHNGRVAUPPRBEMYFREVAUXXT

Table des matières

Résumé	v
Abstract	v
Remerciements	vii
Acronymes	xv
1 Capture du champ lumineux	1
1.1 Rappels sur l'imagerie conventionnelle	1
1.2 Champ lumineux et fonction plénoptique	2
1.3 Système de capture du champ lumineux	4
1.3.1 Capteur plénoptique défocalisé (type 1)	4
1.3.2 Capteur plénoptique focalisé (type 2)	7
1.3.3 Matrice de caméras	7
1.4 Capteur plénoptique intégré de type 1 en détail	8
1.5 Positionnement et plan	10
2 Simulations électromagnétiques	13
2.1 Microlentilles centrées	14
2.1.1 Présentation des simulations et du quad-pixel	14
2.1.2 Critère de performance	15
2.1.3 Optimisation d'un quad-pixel 3.5 μm	15
2.1.4 Résultats complets	17
2.2 Microlentilles décentrées	19
2.2.1 Rappels sur Chief Ray Angle	19
2.2.2 Méthode d'optimisation du CRA	20
2.2.3 Résultats en CRA	24
2.2.4 Analyse en longueur d'onde	24
2.3 Comparaison avec l'état de l'art	25
2.3.1 Dual pixel	25
2.3.2 Impact de la taille de la microlentille	26
2.4 Pistes d'améliorations des performances	28
2.4.1 Diminution du crosstalk inter microlentilles	29
2.4.2 Diminution du crosstalk intra microlentilles	30
2.5 Conclusion	32

3	Rendu en lancé de rayons	33
3.1	Différentes approches pour le rendu d'images	33
3.2	Modélisation de la fonction de flou	34
3.2.1	Spot diagramme et champ de lumière	34
3.2.2	Modélisation de la fonction de flou par image de sous-ouverture	35
3.3	Simulations croisées et rendu d'images	37
3.3.1	Présentation de l'outil de rendu existant	37
3.3.2	Méthode d'intégration des réponses angulaires	38
3.3.3	Validation	40
3.4	Application à un capteur quad-pixel	41
3.4.1	Exemple d'images générées avec PBRT modifié	41
3.4.2	Impact de la diffraction sur les SAIs	43
3.5	Conclusion	43
4	Traitement de l'image	45
4.1	Base de données d'images	46
4.2	Rappels d'éléments en traitement d'images	47
4.2.1	Rappels sur l'algorithme de <i>block-matching</i>	47
4.2.2	Fonctions de coût	47
4.2.3	Précision sous-pixellique	48
4.2.4	Transformation locale des pixels	48
4.2.5	Limite du block matching dans le cas du quad-pixel	50
4.2.6	Convolutions discrètes	51
4.2.7	Fonctions d'activation	51
4.3	Correction des aberrations	52
4.3.1	Principe	52
4.3.2	Comparaison des performances selon le niveau de diffraction	53
4.4	Apprentissage profond	55
4.4.1	Dématriçage	56
4.4.2	Correction de la diffraction	59
4.4.3	Estimation passive de profondeur	63
4.5	Conclusion	64
5	Expériences	67
5.1	Dual pixel Canon	67
5.2	Prototype de capteur quad-pixel	69
5.2.1	Capteur	69
5.2.2	Processus de fabrication	70
5.2.3	Banc de mesure en angle	72
5.2.4	Traitement des données	73
5.2.5	Facteurs d'erreurs	75
5.2.6	Réponses angulaires	76
5.2.7	Images	82
5.3	Conclusion	84

6 Conclusion et perspectives	87
6.1 Résumé	87
6.1.1 Manuscrit	87
6.1.2 Publications	87
6.2 Poursuites et ouvertures	88
Bibliographie	99
Liste des figures	99
Liste des tables	105

Acronymes

BSIBack Side Illuminated
CRAChief Ray Angle
DTIDeep Trench Isolation
FCNFully Convolutional Networks
FDTDFinite Difference Time Domain
FPNFixed Pattern Noise
FSIFront Side Illuminated
MADMean of Absolute Differences
MSEMean Squared Error
PBRTPhysically Based Rendering Theory
PSFPoint Spread Function
PSNRPeak Signal to Noise Ratio
SADSum of Absolute Differences
SAISub Aperture Images
SSDSum of Squared Differences
ZNCCZero means Normalized Cross Correlation

Capture du champ lumineux

1.1 Rappels sur l'imagerie conventionnelle

La photographie numérique offre la possibilité d'enregistrer un sujet ou une scène sous la forme d'une image fixe, facilement stockable, partageable et affichable. Elle se base sur les développements des technologies de photodiodes CCD[25, 71] puis CMOS[7, 24], qui permettent de transformer la lumière en signal électrique, mais aussi sur les progrès de l'informatique logicielle et matérielle (format de représentation et de compression des données, périphériques de stockage et d'affichage...). Trois éléments composent le système d'imagerie : l'objectif, le capteur et enfin le traitement de l'image (Figure 1.1).

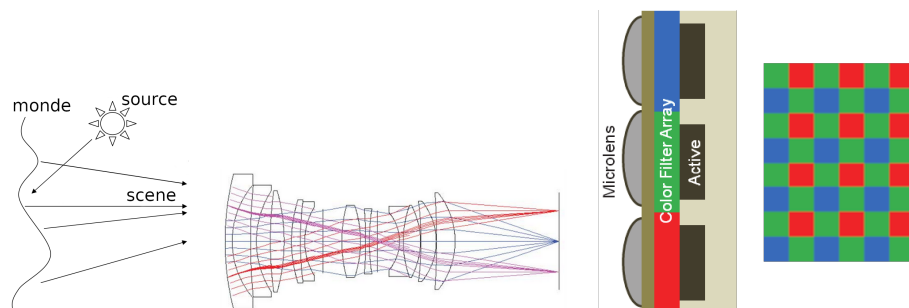


Figure 1.1. Chaîne d'imagerie conventionnelle : la scène renvoie des rayons de la source lumineuse vers l'objectif. Ce dernier forme une image sur le capteur qui l'échantillonne et l'enregistre. Les données enregistrées sont traitées informatiquement pour créer l'image finale.

En présence d'une source de lumière, un objet va renvoyer les rayons lumineux dans tout l'espace. Un objet réfléchissant comme un miroir va les renvoyer dans une direction privilégiée, tandis qu'un objet dit *lambertien* va les renvoyer dans toutes les directions, y compris vers l'objectif. Le but de celui-ci est de former une image nette de la scène, en faisant converger les rayons issus d'un même point de la scène au même endroit sur le capteur. Il est caractérisé par sa longueur focale et son ouverture numérique ($F\# = \text{focale} / \text{diamètre}$). La longueur focale est la distance à laquelle l'image d'un objet à l'infini se forme, plus elle est longue plus l'angle de vue est réduit ("*télé-objectif*"), plus elle est courte plus l'angle de vue est grand ("*grand angle*"). L'ouverture, en simplifiant, décrit

la quantité de lumière que l'objectif laisse passer. Plus le nombre $F\#$ est petit, plus l'objectif laisse passer de lumière. On cherche idéalement à avoir une grande ouverture, pour permettre la prise de vue même en conditions de luminosité difficiles (obscurité, mauvais éclairage).

Le capteur enregistre l'image produite par l'objectif en convertissant l'intensité lumineuse en signal électrique. Cette conversion est gérée par des photodiodes régulièrement espacées sur le capteur. L'image est donc spatialement échantillonnée. Pour former une image en couleur, on place des filtres colorés au-dessus des photodiodes. Ainsi une photodiode voit seulement une partie du spectre lumineux. Le motif le plus souvent utilisé est celui de Bayer où l'on trouve en alternance des filtres rouges, verts et bleus. Le signal électrique est converti en valeur numérique puis stocké dans un fichier. Le terme pixel en traitement d'image fait référence à un triplet (R,V,B) à une certaine coordonnée de l'image. Utilisé dans le cadre du capteur, il fait plutôt référence à l'ensemble formé par la photodiode et son circuit de lecture.

Pour produire l'image finale, il faut traiter les données enregistrées. Des étapes "*bas-niveau*", c'est-à-dire proches de la physique de la chaîne d'acquisition, sont d'abord nécessaires. On peut citer principalement le dématriçage couleur[51] et la réduction du bruit[35]. Chaque photodiode voyant une seule couleur, le dématriçage consiste à trouver les deux autres couleurs restantes en fonction de la valeur des pixels adjacents et au final former le triplet (R,V,B) complet. La réduction du bruit est aussi une étape nécessaire car plusieurs phénomènes peuvent faire fluctuer aléatoirement la valeur enregistrée par pixel : la nature discrète de la lumière, l'agitation thermique, la conversion analogique/numérique, etc... Une fois ces étapes réalisées, on peut trouver des traitements de plus *haut niveau* (plus proches de la vision humaine) tels que l'ajustement de l'exposition, du contraste, ou la gestion de la balance des blancs, jusqu'à trouver des méthodes de reconnaissance d'objets, de visages ou même des panneaux de signalisation dans le cas de la conduite autonome.

1.2 Champ lumineux et fonction plénoptique

Comme on vient de le voir, la photographie permet de capturer une scène, c'est-à-dire d'enregistrer l'intensité des rayons lumineux issus des différents objets. Le champ de lumière (*light-field* en anglais) représente l'ensemble des rayons au sein d'une scène. Cette notion a été conceptualisée par Gershun en 1939[33]. Il représente les intensités, ou radiances (en W/m^2), de l'intégralité des rayons lumineux d'une scène, en faisant l'hypothèse que l'intensité est constante le long du rayon en l'absence d'obstacle. La fonction plénoptique, introduite par Adelson en 1991[3], décrit le champ lumineux à l'aide de sept paramètres :

$$L = L(x, y, z, \theta, \phi, \lambda, t) \quad (1.1)$$

où (x, y, z) sont les coordonnées d'un point dans l'espace par lequel passe un rayon avec un angle (θ, ϕ) et une longueur d'onde λ à un moment t . Si l'on étudie une seule longueur d'onde et que l'on considère une scène figée dans le temps, on peut supprimer les deux dernières variables, ce qui donne une fonction L à cinq variables, comme illustré sur la Figure 1.2 a).

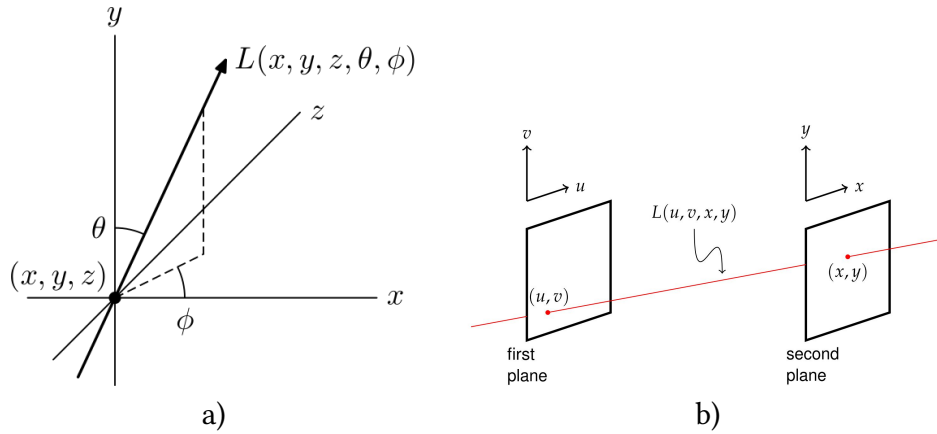


Figure 1.2. Représentation d'un rayon avec cinq coordonnées à gauche, puis avec une paramétrisation à deux plans à droite. Illustration tirée de Lam 2015[43].

Levoy simplifie cette fonction en 1996[46] en proposant une paramétrisation à deux plans (x, y, u, v) (Figure 1.2 b). A partir de cette paramétrisation, on peut encore réduire l'espace de travail en considérant qu'une coordonnée dans un plan est indépendante de l'autre coordonnée. Cela permet un découplage entre u et v , et x et y . On définit alors un diagramme dit rayon-espace (Figure 1.4) où l'on représente les points d'origine des rayons sur l'axe u en fonction de leur point d'impact sur l'axe x . Une caméra standard munie d'un objectif standard projette le champ lumineux dans un espace à deux dimensions, l'information de direction des rayons est donc perdue (Figure 1.3). On doit à Lippmann en 1908[49] puis Adelson en 1912[4] une approche théorique pour capturer la direction des rayons lumineux. Il a fallu attendre les années 90 et le développement à la fois des moyens informatiques et des capteurs d'images pour pouvoir envisager réellement la capture du champ lumineux.

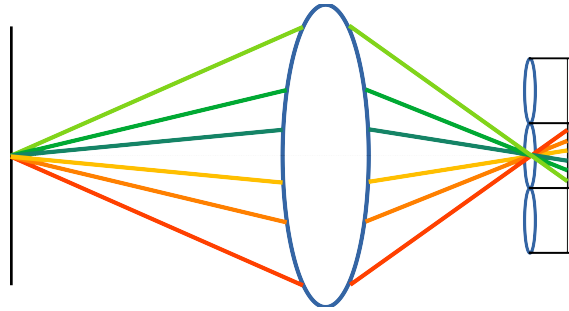


Figure 1.3. Capture du champ de lumière par la chaîne d'acquisition classique : l'information de direction est perdue. Les rayons sont intégrés de la même manière quel que soit leur angle d'incidence.

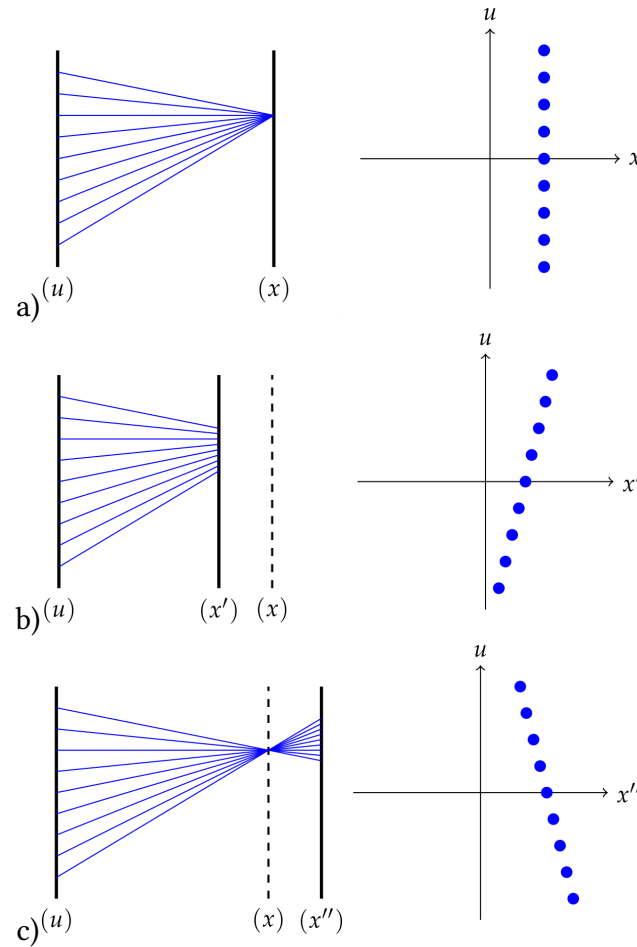


Figure 1.4. Exemples de diagrammes rayon/espace. La figure a) représente le cas où les rayons convergent en un même point sur l'axe x quelles que soient leurs coordonnées d'origine sur l'axe u . Les cas b) et c) représentent les cas où les rayons convergent devant et derrière le plan (x, y) considéré. Illustration tirée de Lam 2015[43].

1.3 Système de capture du champ lumineux

La capture du champ lumineux consiste à non seulement enregistrer l'intensité des rayons mais aussi leur direction. Pour simplifier, au lieu de produire une image fixe en deux dimensions, on peut arriver à recréer la scène en trois dimensions (en tout cas en partie, selon la méthode choisie).

1.3.1 Capteur plénoptique défocalisé (type 1)

Une manière de capturer un champ lumineux, popularisée en 2006 par Ren Ng [54, 57, 56] et étudiée par Georgiev[31] ou Levoy[45], est de placer une grille de microlentilles devant un capteur standard. La distance entre les microlentilles et le capteur correspond à leur distance focale. Un tel capteur est appelé un capteur de type 1, ou défocalisé. Les microlentilles correspondent à un échantillonnage spatial et ont pour fonction de diriger les rayons sur différents pixels selon leur angle d'incidence, ce qui produit l'échantillonnage angulaire. La Figure 1.6 illustre le principe. Lorsque l'objectif produit une image nette

sur les microlentilles, celles-ci forment une tâche de flou uniforme. On comprend que ce système sacrifie la résolution spatiale pour la résolution angulaire : lorsqu'un objet est au point, les pixels n'apportent pas d'information supplémentaire.



Figure 1.5. Une Lytro illum.

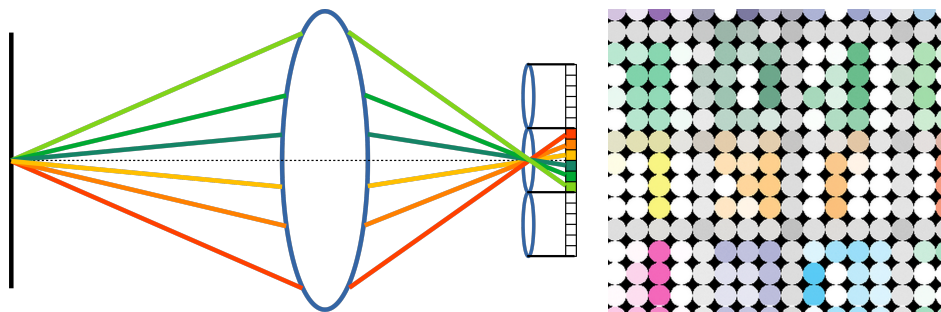


Figure 1.6. Capteur plénoptique de type 1 (dit défocalisé). Ici les rayons sont intégrés par des pixels différents en fonction de leur angle d'incidence.

La grille de microlentilles était collée sur le capteur, ce qui dégradait la résolution des images refocalisées ou la précision des cartes de profondeur à cause des défauts d'alignement dus au collage. Cela amenait aussi la nécessité de procéder à un calibrage coûteux, caméra par caméra, pour estimer les décalages des microlentilles. La possibilité de changer la mise au point en post-production sur tout le champ a été mise en avant mais s'est avérée trop coûteuse en résolution spatiale : les produits Lytro (Figure 1.5) n'ont pas trouvé leur public.

En parallèle, les constructeurs de capteurs d'images comme Canon, Samsung ou Sony ont développé des capteurs plénoptiques intégrés de type 1, dits "dual-pixels", où une microlentille couvre deux sous-pixels rectangulaires (Figure 1.7), principalement pour contrôler l'autofocus de leurs appareils, en photo comme en vidéo. Cet autofocus est plus rapide et précis que la mesure de contraste car on peut savoir dans quel sens déplacer la mise au point, là où la mesure par maximisation de contraste doit chercher à l'aveugle. Ces entreprises savent former directement les microlentilles sur les capteurs pendant leur fabrication avec des technologies de micro-électronique avancées. Ces derniers ne souffrent pas d'une imprécision de positionnement des microlentilles car leur placement se fait à quelques dizaines de nanomètres près. Le premier produit de ce type est le reflex 70D de Canon, sorti en 2013. Le logiciel propriétaire de Canon permet d'en tirer partie en offrant la possibilité d'améliorer le flou d'arrière, de réduire certains effets parasites (ghosting) ou

de refaire le point après la prise de vue. Cette possibilité est plus réduite qu'avec une caméra Lytro mais permet par exemple de rattraper une erreur de l'autofocus en changeant le plan de netteté du nez à l'œil[83].



Figure 1.7. Un Canon 70D et une vue de ses sous-pixels rectangulaires. La partie en niveaux de gris est une photo prise avec un microscope électronique à balayage où l'on voit deux pixels et les transistors formant leur circuit de lecture. La vue de droite est issue d'un microscope à force atomique capacitif où l'on voit deux pixels entiers, formés par deux sous-photodiodes rectangulaires (rectangles en doré notés "PD"). Les microlentilles ne sont pas visibles, elles sont placées au dessus des groupes de photodiodes.

La technologie dual-pixel a été reprise dans les capteurs de smartphones et Google a commencé à s'en servir pour des usages plus avancés que l'autofocus. Le mode portrait se base sur la mesure de profondeur passive pour segmenter le premier plan et l'arrière-plan (Figure 1.8) et ainsi créer un flou d'arrière-plan artificiel[81, 27, 88]. On peut également citer d'autres cas d'usage comme l'extension de l'angle de vue[78], la super-résolution[79] ou les corrections de réflexions parasites[63]. Contrairement aux caméras "entièrement" dédiées à la capture du champ lumineux (Lytro), ce type de capteurs intégrés se développent de plus en plus. On commence à voir apparaître des brevets Canon et Sony concernant les quad-pixels (2x2 sous-pixels carrés par microlentille) et même des nona-pixels (3x3) chez Samsung.

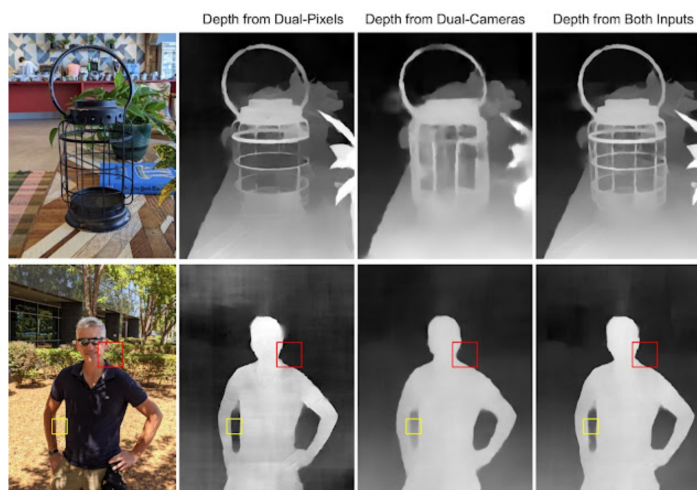


Figure 1.8. Carte de profondeur générée par Google à l'aide d'un capteur dual-pixel et d'un système dual-caméras. La colonne la plus à droite est la fusion des deux systèmes.

1.3.2 Capteur plénoptique focalisé (type 2)

Lumsdaine[52] et Georgiev[32, 30] ont proposé une caméra où les microlentilles ne sont plus placées à leur distance focale, mais de telle sorte que l'objectif focalise dans un plan placé devant ou derrière les microlentilles (Figure 1.10). On forme alors une multitude d'images sur le capteur, chacune offrant un point de vue légèrement décalé par rapport à ses voisines. Ce capteur est appelé un capteur de type 2, ou focalisé. Dans ce cas-là, la perte en résolution spatiale est réduite[28]. C'est ce type de caméra qu'utilise l'entreprise Raytrix (Figure 1.9) pour des applications de vision industrielle ou scientifique. De plus, ils utilisent trois microlentilles de focales différentes sous forme de matrice hexagonale pour étendre la carte de profondeur. On peut noter que le marché de Raytrix étant plus ciblé, c'est la seule entreprise à concevoir et vendre des caméras plénoptiques actuellement. Des applications ont été montrées pour la microscopie[47] ou la super-résolution[29].

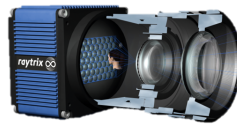


Figure 1.9. Une camera Raytrix.

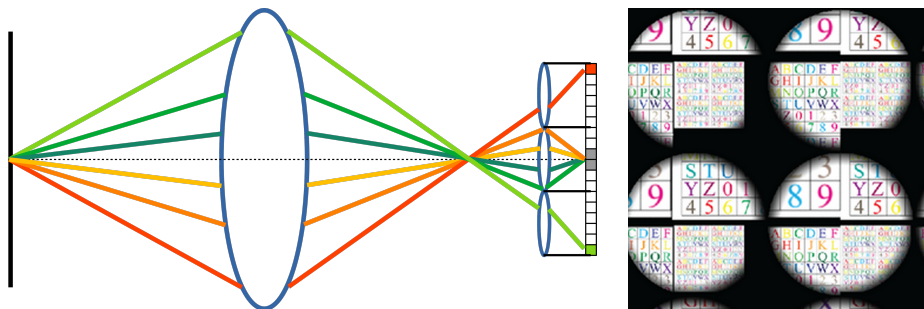


Figure 1.10. Capteur plénoptique de type 2 (dit focalisé).

1.3.3 Matrice de caméras

On trouve enfin le système multi-caméras[82] utilisé par l'entreprise Pelican Imaging par exemple, mais aussi étudié par l'équipe light-field d'Interdigital. L'échantillonnage en angle se fait cette fois en plaçant plusieurs caméras séparées d'une certaine distance (la *baseline*). Les tentatives d'intégrer ce genre de concept ont aussi échoué, par exemple les produits de Pelican ou le smartphone L16 et ses 16 modules photos. Cependant les matrices de caméras de grande taille continuent de se développer. Le but est de créer des contenus 3D immersifs (orientés loisirs et multimédia), dans ce cas la capture du champ lumineux est utilisé principalement pour l'estimation de la profondeur, les applications comme la refocalisation ne sont pas utilisées. Des travaux sont en cours, entre autres, sur le dématricage du champ lumineux (Yu[86], Seifi[69]) ou la synthèse de vues intermédiaires (Srinivasan[72], Hog[39], Navarro[53]).



Figure 1.11. Dans l'ordre : une caméra Pelican et la matrice de caméras d'Interdigital.

1.4 Capteur plénoptique intégré de type 1 en détail

Cette thèse porte sur l'étude d'un capteur plénoptique intégré quad-pixel de type 1, c'est-à-dire où l'on place une microlentille au-dessus de 2×2 pixels carrés et dont la distance focale correspond à la distance au pixel. Dans cette partie, je détaille les différents concepts utilisés pour traiter les données issues d'un tel capteur.

1.4.0.1 Images de sous-ouverture (Sub Aperture Images)

Une manière classique d'utiliser un capteur plénoptique de type 1 est d'extraire les images de sous-ouvertures (SAI). Elles correspondent aux images formées par les sous-pixels ayant la même position sous les microlentilles, telles que montrées sur la Figure 1.12. Soit p la taille plénoptique (le nombre de sous-pixels sous une microlentille dans une direction), (x, y) les coordonnées des sous-pixels sur le capteur et (u, v) les coordonnées des sous-pixels par rapport aux microlentilles. Si l'on appelle (α, β) les coordonnées des pixels dans une image de sous-ouverture, on a :

$$(\alpha, \beta, u, v) = \left(\left\lfloor \frac{x}{p} \right\rfloor, \left\lfloor \frac{y}{p} \right\rfloor, x \bmod p, y \bmod p \right) \quad (1.2)$$

avec $\lfloor . \rfloor$ désignant la fonction partie entière et \bmod la fonction modulo.

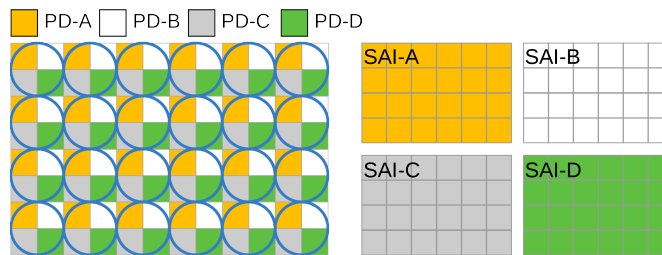


Figure 1.12. Définition des images de sous-ouverture (Sub Aperture Images).

L'image finale est la somme des images de sous-ouverture.

1.4.0.2 Disparités

Selon la mise au point de l'objectif utilisé et la distance des objets de la scène, des translations locales entre les SAIs peuvent apparaître. On appelle ces translations les disparités, notées ρ . Elles sont illustrées par la Figure 1.13.

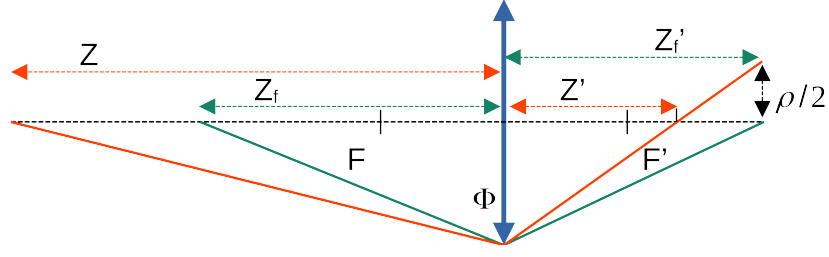


Figure 1.13. Illustration de la disparité.

On considère une lentille mince de diamètre Φ et de focale f . Z_f représente la distance de mise au point et Z la distance de l'objet. On utilise l'équation des lentilles minces puis le théorème de Thalès après la lentille pour trouver :

$$Z'_f = \frac{f \times Z_f}{Z_f - f} \text{ et } Z' = \frac{f \times Z}{Z - f} \quad (1.3)$$

$$\rho = \Phi \left(\frac{Z'_f - Z'}{Z'} \right) \quad (1.4)$$

En combinant ces équations on trouve :

$$\rho = \frac{\Phi \times Z_{focus} \times f}{Z_{focus} - f} \times \left(\frac{1}{Z_{focus}} - \frac{1}{Z} \right) \quad (1.5)$$

La disparité est une valeur en mètres pouvant être positive ou négative selon la position de l'objet et la distance de mise au point. C'est là-dessus que se base l'autofocus des capteurs dual-pixels par exemple, cela permet de savoir dans quel sens faire varier la mise au point pour obtenir un objet net. Il est possible d'obtenir une disparité en pixels en divisant la définition précédente par le diamètre d'une microlentille.

1.4.0.3 Refocalisation

Les SAIs étant extraites, il est possible de refocaliser l'image finale en suivant l'équation suivante :

$$\begin{bmatrix} X \\ Y \end{bmatrix} = s \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + s \times \rho_{focus} \begin{bmatrix} u \\ v \end{bmatrix} \quad (1.6)$$

où (X, Y) sont les nouvelles coordonnées des pixels, (x, y) sont les anciennes, s est un paramètre de dilatation (la plupart du temps laissé à 1 dans cette thèse) et ρ_{focus} la disparité pour laquelle un objet à une distance d_{focus} est net. Cela revient à les translater les unes par rapport aux autres pour ramener les pixels issus d'un même objet à la même coordonnée image (Figure 1.14). Si on reprend les diagrammes rayons-espace de la Figure 1.4, cela revient à ré-aligner les rayons des cas b) ou c) pour se ramener au cas a). Un exemple de refocalisation de l'arrière-plan et du premier-plan est montré sur la Figure 1.15.

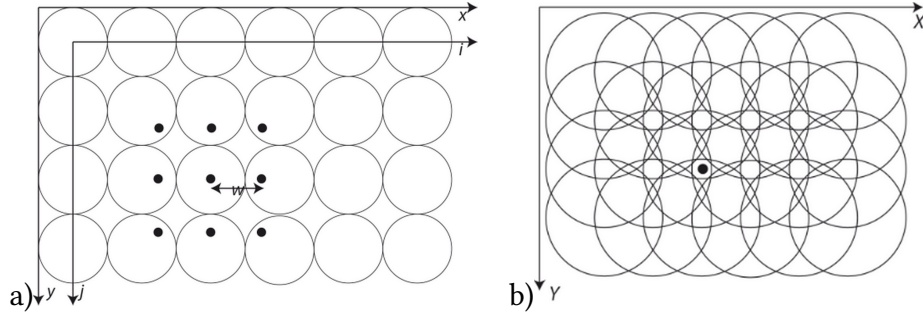


Figure 1.14. Illustration du principe de la refocalisation.

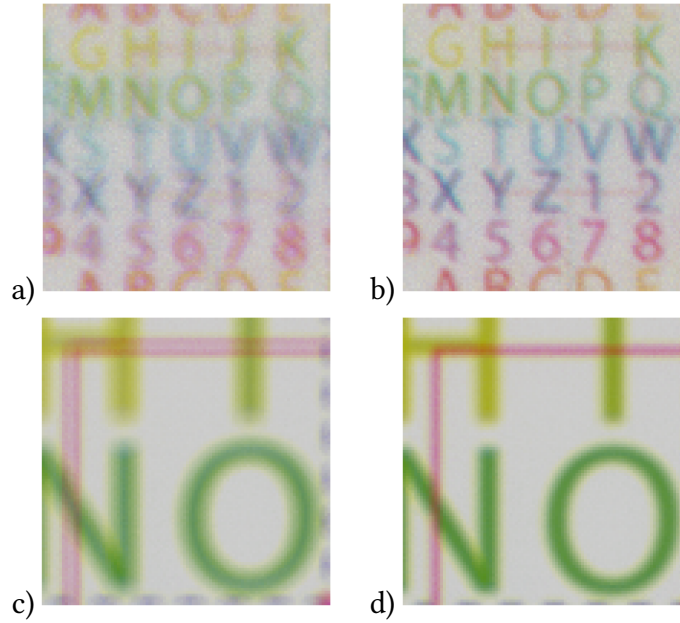


Figure 1.15. Exemple de refocalisation d'une image. L'image a) est une mire de texte placée au-delà du plan de netteté, sans refocalisation, donc floue. L'image b) est la version refocalisée avec un paramètre $\rho = +2$, qui est maintenant nette. Les images c) et d) illustrent le cas où la mire est placée devant le plan de netteté et où l'on a utilisé un paramètre $\rho = -2$.

1.5 Positionnement et plan

Il apparaît que les fabricants de capteurs photographiques tels que Canon, Samsung ou Sony concentrent leurs efforts sur l'autofocus, tandis que Google pousse sur le développement du traitement d'image. Cette thèse porte sur les capteurs quad-pixels où une microlentille recouvre un patch de 2×2 sous-pixels. Elle a pour objectif de tisser des liens entre deux des trois aspects du système d'imagerie, les quad-pixels d'une part, et le traitement de l'image d'autre part. En particulier, on cherche à connaître l'impact de l'architecture du pixel sur le traitement de l'image. Un autre aspect est de développer des algorithmes dédiés et optimisés pour le quad-pixel. L'objectif utilisé pour produire l'image sur le capteur est quant à lui laissé de côté (on peut citer les travaux de Trouvé[76] ou Burcklen[10, 11, 12] sur le sujet, où la co-conception est faite entre l'optique et le trai-

tement d'image). L'axe de co-conception entre l'optique et le capteur ne sera pas abordé non plus.

La thèse a été menée à la fois au CEA-Leti (Grenoble), dans le laboratoire d'imagerie sur silicium sous la direction de Jérôme Vaillant, et dans l'équipe light-field d'Interdigital R&D France (anciennement Technicolor R&I, Rennes) avec l'encadrement de Benoit Vandame. Le LIS a apporté son savoir faire sur la conception, les simulations et la fabrication de pixels dans le domaine du visible, et l'équipe light-field a apporté ses compétences sur l'acquisition du champ de lumière et les algorithmes associés.

Le manuscrit est organisé de la façon suivante :

- Dans le chapitre 2, je mène des simulations au niveau des pixels, principalement en utilisant des sous-pixels de $1.75\mu\text{m}$. Ce travail fait suite aux simulations en dual-pixel menées par Canon et Samsung. A partir du critère de performance défini, je propose à la fin du chapitre des pistes d'amélioration des pixels.
- Dans le chapitre 3, je présente une méthode pour prendre en compte les simulations des pixels dans un logiciel de rendu en lancé de rayons. Cela a permis de générer une base de données d'images sur lesquelles on s'appuie pour étudier les procédures de traitement d'images.
- Dans le chapitre 4, j'étudie l'impact de la diffraction par les microlentilles sur un algorithme de correction d'aberrations. J'utilise ensuite des réseaux de neurones pour aborder d'autres problèmes comme le dématricage couleur ou l'estimation de profondeur.
- Enfin dans le chapitre 5, je décris et caractérise un prototype quad-pixel réalisé dans le cadre de cette thèse. Il est comparé aux simulations et utilisé pour prendre quelques images.

Simulations électromagnétiques

Un capteur plénoptique de type 1 est réalisé lorsqu'on place plusieurs sous-pixels sous une même microlentille, telle que la distance entre celle-ci et le silicium soit égale à sa focale. Une façon d'y parvenir est d'aligner puis de coller une grille de grandes microlentilles, indépendante, sur un capteur CMOS standard. Cette technique permet de placer un grand nombre de sous-pixels par microlentilles, et donc d'obtenir de très bonnes capacités en traitement d'images plénoptiques, mais au prix de la résolution spatiale. De plus, l'alignement n'est pas parfait et peut varier entre $1\text{ }\mu\text{m}$ et $10\text{ }\mu\text{m}$ selon l'équipement utilisé. Il est donc nécessaire de calibrer chaque capteur en post-traitement, ce qui peut se révéler difficile et/ou onéreux. La technique alternative est de faire réaliser ces microlentilles directement par le fabricant du capteur. Les procédés de fabrication limitent le diamètre et la distance au silicium des microlentilles et donc limitent le nombre de sous-pixels, mais cette approche est parfaitement adaptée au quad-pixel où l'on place seulement 2×2 sous-pixels par microlentille. Dans ce cas, aucune calibration n'est nécessaire car le placement des microlentilles peut s'effectuer à quelques dizaines de nanomètres près. Cependant une limitation évidente est que l'on perd en capacités plénoptiques, par exemple en refocalisation ou en estimation de profondeur. Une autre limitation vient de la taille de ces microlentilles. Contrairement à la grille de microlentilles où l'on peut considérer l'approximation des rayons lumineux valide et travailler avec les règles de l'optique géométrique, on ne peut plus négliger la diffraction à l'échelle du pixel. Il faut alors optimiser plus finement le quad-pixel, en particulier le rayon de courbure et la distance microlentille-silicium. Des simulations électromagnétiques ont été menées par Canon et Samsung dans le cas du dual-pixel [42, 19, 87]. Des simulations similaires concernant le quad-pixel ont été menées par Sony[58], Canon[26] et Samsung[64] récemment.

C'est aussi ce que j'étudie et approfondis dans ce chapitre. Je décris d'abord le modèle de pixel qui sera simulé à l'aide d'un logiciel de Différences Finies dans le Domaine Temporel[37] (FDTD), ainsi que le critère qui permet de juger les performances. Dans le cas d'une microlentille centrée, je m'intéresse à deux tailles de sous-pixels, $1.75\text{ }\mu\text{m}$ et $0.8\text{ }\mu\text{m}$. Dans le cas d'une microlentille décentrée pour corriger l'angle du rayon principal, je décris la méthode d'optimisation et de placement pour des sous-pixels de $1.75\text{ }\mu\text{m}$ uniquement. Les réponses angulaires complètes sont données à la suite. Je compare ensuite mes travaux à l'état de l'art existant, en particulier à Canon et Samsung et leurs capteurs dual-pixels, et j'étudie l'impact de la taille des microlentilles sur les performances. Enfin, je propose dans la dernière section des architectures de quad-pixels moins standards qui ont pour but de diminuer les sources de crosstalk.

Outre l'aspect théorique, mes simulations permettent de préparer le prototype réalisé à Grenoble par le CEA. En effet, des pixels de $1.75\ \mu\text{m}$ correspondent à un capteur de test déjà existant et les microlentilles de $3.5\ \mu\text{m}$ sont elles aussi relativement faciles à fabriquer.

2.1 Microlentilles centrées

2.1.1 Présentation des simulations et du quad-pixel

Le but des simulations est de calculer la puissance absorbée par le silicium de chaque sous-pixel étant donné l'angle d'incidence de la lumière. On la note $P_{\text{abs}}^{\text{id}}$ avec id l'indice du sous-pixel, dans notre cas A,B,C ou D. La lumière incidente est une onde plane polarisée linéairement et de longueur d'onde $\lambda=550\ \text{nm}$ (sauf indication contraire). Son angle d'incidence est décomposé selon φ la coordonnée polaire et θ la coordonnée azimutale ($0 \leq \theta; 0 \leq \varphi \leq 360$). Pour chaque angle d'incidence on simule en fait deux polarisations orthogonales (0° et 90°) et $P_{\text{abs}}^{\text{id}}$ est la moyenne arithmétique des deux simulations. Dans le logiciel de simulation utilisé, on utilise un minimum de 10 cellules de Yee par longueur d'onde. Les conditions aux frontières sur l'axe z (normale à la surface du silicium) sont absorbantes : cela permet d'éviter de simuler le champ électrique qui est réfléchi par la microlentille et celui qui n'est pas totalement absorbé par le silicium. Les conditions aux frontières sur les axes x et y sont périodiques : cela permet de simuler un potentiel crosstalk entre les microlentilles lorsque l'angle d'incidence devient grand.

Notre modèle de simulation, illustré Figure 2.1, est basé sur un pixel éclairé par la face arrière (BSI, Back Side Illuminated), ce qui permet d'augmenter la surface sensible en retirant les métaux du circuit de lecture. On vient placer plusieurs couches de quelques centaines de nanomètres entre la microlentille et le silicium. Bien qu'en pratique on puisse trouver une multitude de couches selon les procédés de fabrication, on se limite ici à trois couches principales. La couche de planarisation permet l'accroche de la microlentille, la résine colorée filtre les longueurs d'onde (en général rouge, vert, bleu en imagerie visible) et enfin la couche de passivation isole le silicium et améliore la collecte des charges. Le silicium, de quelques microns d'épaisseur, est divisé en quatre zones qui correspondent aux quatre sous-pixels par des tranchées d'isolation[5] (DTI, Deep Trench Isolation) de $100\ \text{nm}$ de large.

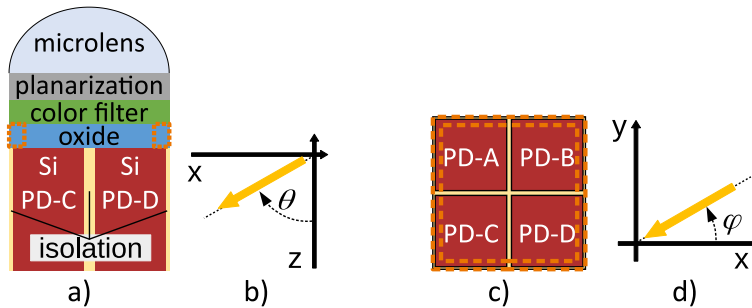


Figure 2.1. Design général d'un quad-pixel dans nos simulations. (a) et (c) montrent la vue de côté et du dessus. (b) et (d) montrent l'orientation des angles θ et φ sur lesquels on décompose l'angle d'incidence de l'onde plane (vue de côté et de dessus respectivement).

2.1.2 Critère de performance

J'ai utilisé le même critère de performance que Kobayashi[42] pour optimiser les quad-pixels. Le principe est de calculer le ratio gauche/droite en fonction de l'angle d'incidence puis de calculer les plages angulaires où le ratio est au dessus d'un seuil arbitraire (R_a) et au-dessous (R_b). Je me suis basé sur un contraste angulaire de 5, c'est-à-dire un ratio de 0.2, pour décider si le signal plénoptique est utilisable.

$$\text{ratio} = \begin{cases} P_{\text{abs}}^{\text{B}}/P_{\text{abs}}^{\text{A}} & \text{si } \varphi = 0 \text{ et } \theta \geq 0 \\ P_{\text{abs}}^{\text{A}}/P_{\text{abs}}^{\text{B}} & \text{si } \varphi = 180 \text{ et } \theta \geq 0 \text{ (équivalent à } \varphi = 0 \text{ et } \theta \leq 0) \end{cases} \quad (2.1)$$

R_a représente la plage angulaire où la diffraction par la microlentille dégrade le contraste entre les sous-pixels. Elle est prédominante lorsque l'angle d'incidence est faible car la tache de diffraction "tombe" sur les quatre sous-pixels en même temps. Cela correspond à un crosstalk entre les sous-pixels, et R_a doit être minimisée. R_b représente la plage angulaire où le contraste angulaire est optimal. Elle est limitée par la diffraction pour des angles d'incidence faibles, et par le crosstalk entre les microlentilles pour des angles d'incidence élevés. R_b doit être maximisée. D'une manière générale :

- L'angle d'incidence maximal avant que le crosstalk entre microlentilles apparaisse doit être maximisé.
- La plage angulaire R_b où le ratio est en dessous d'un certain niveau (ici 0.2) doit être maximisée.
- La plage angulaire R_a où le ratio est au-dessus d'un certain niveau doit être minimisée.
- La puissance totale absorbée $P_{\text{abs}}^{\text{tot}} = \sum P_{\text{abs}}$ doit être maximisée.

Dans le cas du quad-pixel, et d'un capteur plénoptique de type 1 de manière générale, les principaux leviers d'optimisation sont le rayon de courbure et la distance au silicium de la microlentille. Le calcul de la focale de la microlentille à partir de son rayon de courbure (optique géométrique) donne une première approximation mais la diffraction décale le point optimal, d'où la nécessité des simulations **FDTD**. La partie suivante met en application ce critère pour des sous-pixels de $1.75 \mu\text{m}$.

2.1.3 Optimisation d'un quad-pixel $3.5 \mu\text{m}$

Dans cette partie, on étudie un quad-pixel de $3.5 \mu\text{m}$ dont la couche de planarisation en résine transparente fait 500 nm , le filtre coloré en résine verte fait $1 \mu\text{m}$ et la couche de passivation en oxyde de silicium fait 650 nm . On essaye plusieurs valeurs pour le rayon de courbure de la microlentille et pour la hauteur de la microlentille : $2.48 \mu\text{m}$, $3.00 \mu\text{m}$ et $3.5 \mu\text{m}$, ce qui donne neuf combinaisons différentes décrites en Figure 2.2c). L'utilisation de microlentilles jointives, c'est-à-dire de base carrée et sans espace entre elles, contraint la valeur minimale du rayon de courbure à $2.48 \mu\text{m}$ (la diagonale d'un sous-pixel). Ce que l'on appelle "hauteur" de la microlentille est la distance entre la base et son sommet. Dans les cas où la hauteur est plus grande que le rayon de courbure, cela signifie qu'il existe une "base" carrée du même matériau que la microlentille placée entre la surface courbée et la couche de planarisation.

Géométriquement, une microlentille de longue focale éloignée du silicium devrait permettre une meilleure sensibilité en angle, c'est-à-dire diminuer R_a , car la distance au centre en fonction de l'angle d'incidence évolue en $f \times \tan(\theta)$. Elle devrait être cependant plus

sensible au crosstalk entre microlentilles pour la même raison. Le raisonnement inverse s'applique pour une microlentille de faible focale et plus proche du silicium. Un rayon de courbure plus faible permettrait toutefois de réduire la taille de la tache de diffraction, en dehors des considérations géométriques, et donc d'améliorer à la fois R_a et R_b . C'est pourquoi on veut essayer plusieurs combinaisons.

Dans les simulations qui suivent, θ varie de 0° à 40° par pas de 5° et de 40° à 60° par pas de 10° . φ vaut 0° , on se limite dans cette partie au ratio gauche/droite. La Figure 2.2 montre les résultats de simulations pour une microlentille de hauteur $2.48\ \mu\text{m}$ et de rayon de courbure $3.5\ \mu\text{m}$ (Figure 2.2a)) puis $2.48\ \mu\text{m}$ (Figure 2.2b)). Les réponses des sous-pixels C et D ne sont pas représentées car elles sont identiques à celles de A et B par symétrie. Dans ces exemples, on considère que le résultat en b) est meilleur que celui en a). En effet, en a) on a $R_b = 13^\circ$ et $R_a = 16^\circ$, tandis qu'en b) $R_b = 18^\circ$ et $R_a = 10^\circ$.

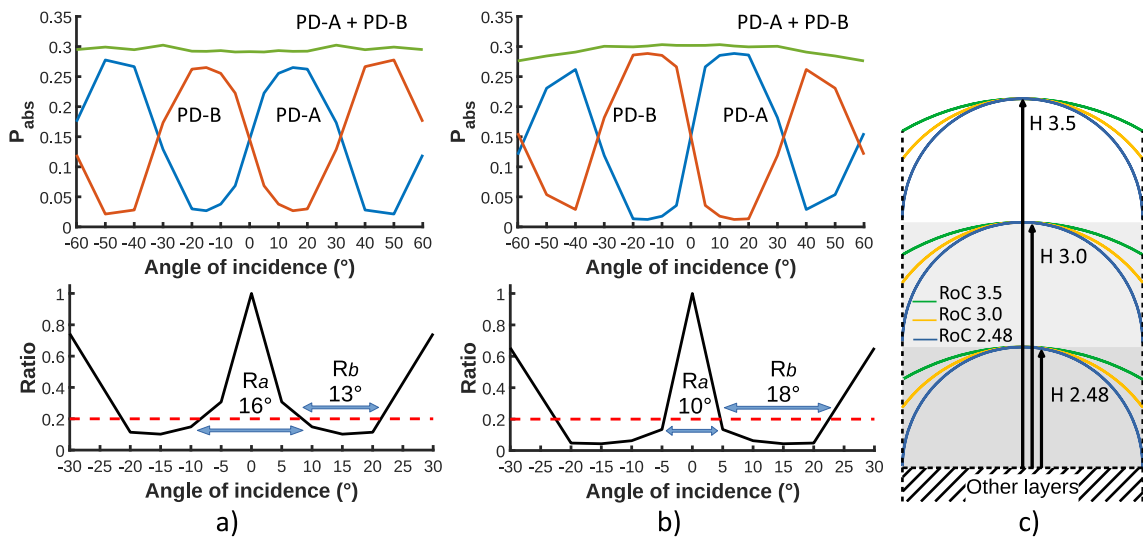


Figure 2.2. Réponse angulaire pour une microlentille de hauteur $2.48\ \mu\text{m}$ et de rayon de courbure de $3.5\ \mu\text{m}$ en a) et de $2.48\ \mu\text{m}$ en b). Les graphiques du haut montrent P_{abs}^A (bleu), P_{abs}^B (orange) et leur somme (vert). Les graphiques du bas représentent le ratio gauche/droite entre -30° et 30° . La figure c) décrit les neuf combinaisons testées.

Sur les neuf jeux de simulations, j'ai déterminé que le design le plus performant est celui où la microlentille est la plus convergente (rayon de courbure à $2.48\ \mu\text{m}$) et la plus proche du silicium (hauteur à $2.48\ \mu\text{m}$). Basé sur ces résultats et étant donné que $2.48\ \mu\text{m}$ est le rayon de courbure minimal, j'ai simulé ce même quad-pixel mais en utilisant un filtre de couleur de $750\ \text{nm}$ au lieu de $1\ \mu\text{m}$ dans le but de réduire encore plus la hauteur totale du pixel. A titre de comparaison, j'ai aussi simulé un quad-pixel de $2.8\ \mu\text{m}$ possédant une isolation tungstène, une "bague" carrée en tungstène placée juste au-dessus du silicium. Elle est représentée par les pointillés oranges sur la Figure 2.1a) et c). On s'attend à ce que la diffraction pour ce pixel plus petit dégrade les performances.

La Figure 2.3a) montre les résultats du pixel de hauteur réduite. Ce design présente des performances légèrement supérieures : R_b augmente de 2° tout en gardant R_a à 10° . La contrepartie de cette stratégie est qu'elle peut dégrader le filtrage en couleur car les résines colorées ont besoin d'une certaine épaisseur pour être efficaces. Il serait aussi possible de diminuer les autres couches au lieu du filtre, comme la passivation ou la planarisation. D'une manière générale ce sont les procédés de fabrication qui limitent l'épaisseur mini-

male de ces couches. Il n'est pas forcément nécessaire de réduire la hauteur au maximum car il faut assez de distance pour que la microlentille fasse converger le flux correctement. Concernant le pixel avec l'isolation tungstène, on remarque que les performances sont bonnes malgré le diamètre réduit de la microlentille. L'isolation permet de gagner 5° sur la plage R_b , cependant le flux total présente un creux à 40° car il est en partie réfléchi par le métal. Ce n'est pas un problème pour les grand pixels car un angle de 40° est difficilement atteignable avec une optique standard, mais cela peut se révéler contre productif avec de petits pixels. En effet, la largeur minimale du tungstène est d'environ 200 nm, or avec des sous-pixels de l'ordre du micron (ou plus petits) on viendrait masquer la surface sensible et finalement perdre l'avantage du BSI.

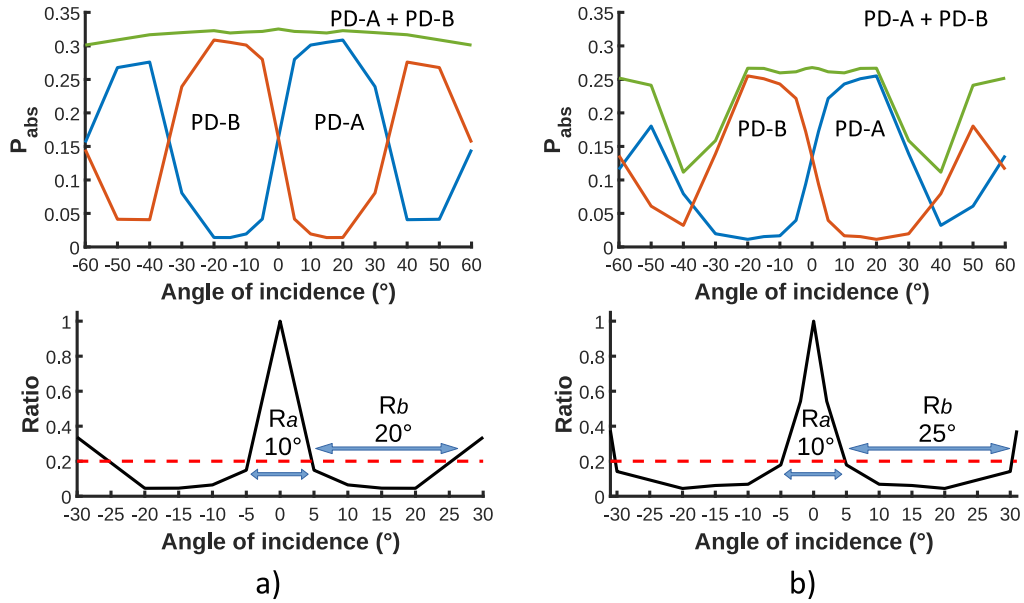


Figure 2.3. Réponse angulaire du pixel 3.5 µm de hauteur réduite a) et du pixel de 2.8 µm avec isolation tungstène b).

2.1.4 Résultats complets

Dans la partie précédente, j'ai travaillé uniquement à $\varphi = 0$, c'est-à-dire en mode dual pixel. Le but du quad-pixel étant d'ajouter l'axe vertical à la mesure des disparités, il est aussi important de connaître sa réponse en deux dimensions. Dans cette partie, je présente les réponses angulaires complètes en φ et θ d'un quad-pixel de 3.5 µm et de 0.8 µm. Ici φ varie de 0° à 330° par pas de 30° . Les différentes couches sont définies dans le Tableau 2.1. Les deux designs présentés ont été optimisés en fonction du critère présenté auparavant, tout en gardant des épaisseurs les plus réalistes possibles. La Figure 2.4 montre les réponses angulaires des quatre sous-pixels pour les quad-pixels de tailles 3.5 µm et 1.6 µm. φ est représenté sur la coordonnée polaire du graphique, θ est représenté sur la coordonnée radiale du graphique. Il est toujours possible de définir un ratio en deux dimensions, mais la définition doit être adaptée pour tenir compte des trois autres sous-pixels. Je propose de diviser la moyenne de trois sous-pixels par le dernier en fonction de φ . La dépendance en φ est similaire à celle du ratio 1D en θ : on calcule droite/gauche si $\theta > 0$ mais gauche/droite si $\theta < 0$. Le ratio 2D s'écrit alors de la manière suivante :

$$\text{ratio} = \begin{cases} \left(P_{\text{abs}}^A + P_{\text{abs}}^B + P_{\text{abs}}^D \right) / 3 \times P_{\text{abs}}^C & 0 \leq \varphi \leq 90 \\ \left(P_{\text{abs}}^A + P_{\text{abs}}^B + P_{\text{abs}}^C \right) / 3 \times P_{\text{abs}}^D & 90 \leq \varphi \leq 180 \\ \left(P_{\text{abs}}^A + P_{\text{abs}}^C + P_{\text{abs}}^D \right) / 3 \times P_{\text{abs}}^B & 180 \leq \varphi \leq 270 \\ \left(P_{\text{abs}}^B + P_{\text{abs}}^C + P_{\text{abs}}^D \right) / 3 \times P_{\text{abs}}^A & 270 \leq \varphi \leq 360 \end{cases} \quad (2.2)$$

Taille du sous-pixel	1.75 μm	0.8 μm
RoC	2.48 μm	1.13 μm
Planarisation	0.5 μm	0.25 μm
Filtre coloré	0.75 μm	0.65 μm
Passivation	0.65 μm	0.2 μm
Silicium	4 μm	4 μm
Largeur des DTI	0.2 μm	0.1 μm

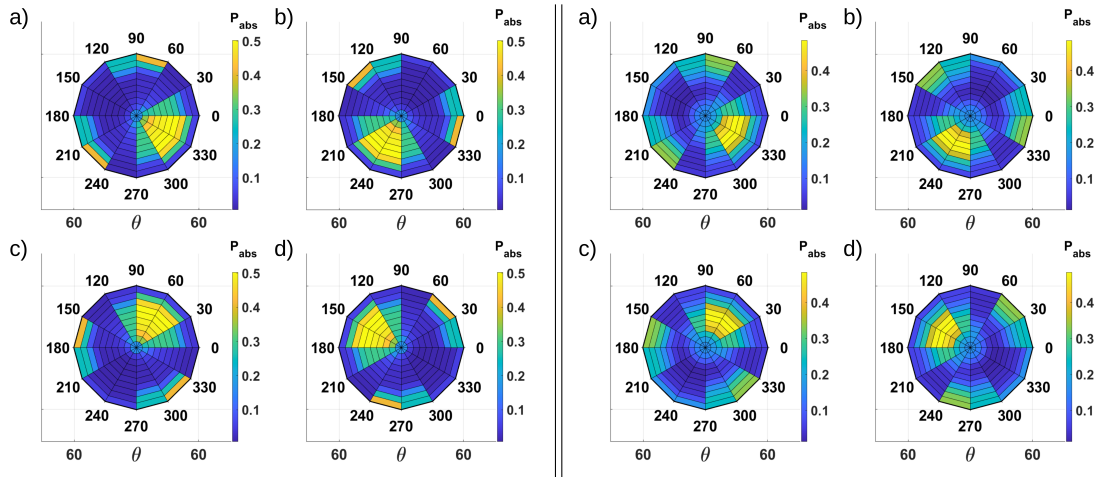
Tableau 2.1. Description des designs des quad-pixels 3.5 μm et 0.8 μm 

Figure 2.4. Réponses angulaires complètes d'un quad-pixel 3.5 μm à gauche et de 1.6 μm à droite. **Note :** il faut faire tourner ces graphes d'un demi-pas en φ (ici 15°) dans le sens horaire pour avoir une visualisation correcte à cause de la manière dont Matlab gère la colorisation du maillage. Cela est valable pour toutes les figures présentant des réponses angulaires en deux dimensions.

Étant donné un seuil arbitraire, l'optimisation porte alors sur la maximisation de la surface du ratio 2D en-dessous de ce seuil. Dans le cas du quad-pixel avec des microlentilles jointives, optimiser sur les axes x ou y en 1D est suffisant. En effet, la diagonale étant plus grande qu'un côté, la plage R_b pour la diagonale est naturellement plus élevée tandis que R_a ne varie quasiment pas. La Figure 2.5 montre les ratio 2D et la surface où ils sont en-dessous de 0.2 pour les deux tailles de quad-pixel étudiés.

Comme on pouvait s'y attendre, la diffraction réduit les performances du quad-pixel de 1.6 μm et en particulier au centre où le contraste angulaire est dégradé sur une plage plus grande. Le crosstalk entre les microlentilles est présent pour des angles d'incidence

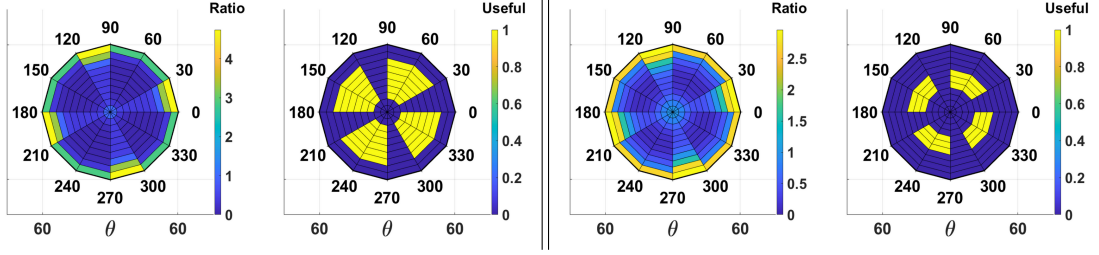


Figure 2.5. Ratio 2D et sa partie en-dessous de 0.2, pour le quad-pixel de $3.5 \mu\text{m}$ à gauche et de $1.6 \mu\text{m}$ à droite.

moins élevés, mais il est quand même possible de travailler jusqu'à environ 25° . Selon les applications et l'objectif utilisé, cela peut suffire.

2.2 Microlentilles décentrées

2.2.1 Rappels sur Chief Ray Angle

Les objectifs utilisés pour l'imagerie grand public ne sont, en général, pas télécentriques dans le plan image (Figure 2.6a), c'est-à-dire que leur pupille de sortie n'est pas à l'infini. C'est d'autant plus vrai dans le cas des smartphones où les lentilles sont très proches du capteur pour des raisons de compacité. On appelle rayon principal le rayon passant par le centre optique du système et n'étant pas dévié par celui-ci. Il forme un angle (Chief Ray Angle, CRA) avec le capteur et le cône de lumière formé par l'optique est centré autour de ce rayon principal (Figure 2.6b). Il est nécessaire de prendre en compte cet angle dans l'optimisation du capteur, sous peine de provoquer du crosstalk entre les microlentilles, mais aussi de décaler la réponse angulaire dans le cadre d'un capteur plénoptique. Il faut absolument éviter ce décalage car on risquerait de perdre le signal plénoptique, par exemple si tout le flux est absorbé dans un seul sous-pixel (Figure 2.6c).

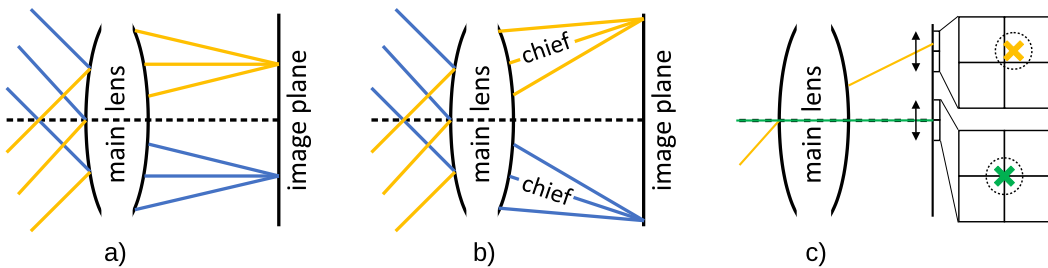


Figure 2.6. (a) et (b) illustrent l'effet d'un objectif télécentrique et non télécentrique. Dans le cas non télécentrique et si les microlentilles sont centrées, la réponse angulaire est décalée.

Pour recentrer la réponse angulaire, la solution usuelle est de décaler les microlentilles par rapport au centre du pixel (Figure 2.7). Ce décalage dépend de la position sur le capteur et de l'objectif utilisé.

Le CRA d'une lentille mince évolue en $\arctan(r/f)$, avec r la distance à l'axe optique dans le plan image, et f la focale. Pour des formules optiques plus complexes, on peut

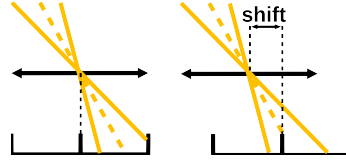


Figure 2.7. Pour un CRA donné, il faut décaler la microlentille pour recentrer la réponse angulaire au centre du pixel.

calculer le CRA à l'aide d'un logiciel de conception optique et on note la fonction obtenue CRA_{func} . La Figure 2.8 en montre deux exemples.

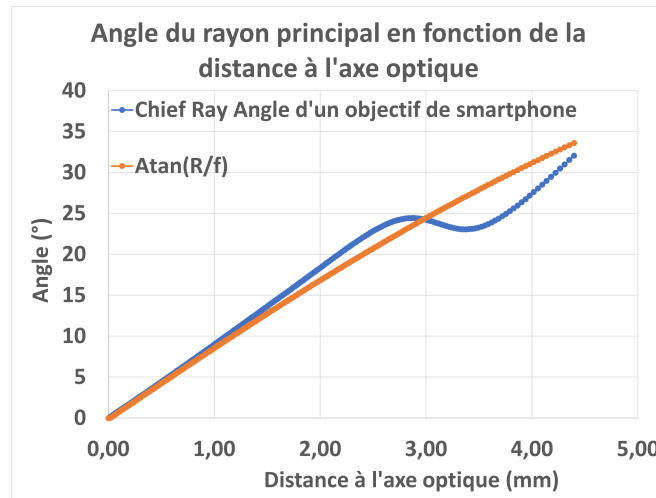


Figure 2.8. Exemple de CRA par rapport à la distance à l'axe optique. La courbe orange est le CRA d'une lentille mince, la bleue celle d'un objectif de smartphone tiré du brevet US8320061-B2.

2.2.2 Méthode d'optimisation du CRA

Dans cette section, je présente la méthode d'optimisation que j'ai utilisée pour calculer le décalage des microlentilles. Pour rester le plus général possible, je ne me suis fixé ni une optique ni une taille de capteur et j'ai encore une fois travaillé en angle, ici φ_{cra} et θ_{cra} . θ_{cra} varie entre 5° et 30° par pas de 5° et φ_{cra} varie entre 0° et 45° par pas de 11.25° . Ce maillage permet de limiter le nombre de simulations requises, mais est aussi et surtout suffisant pour décrire les décalages nécessaires pour tous les φ_{cra} de 0° à 360° . En effet, on peut facilement compléter les points manquants en utilisant les symétries en x et y du quad-pixel.

Une première étape (potentiellement suffisante à elle seule selon le degré de précision requis) est de lancer une onde plane, dont l'angle d'incidence correspond au CRA qu'on cherche à corriger, sur le quad-pixel. On monitore ensuite la tâche de diffraction à la surface du silicium (Figure 2.9) et enfin on calcule son barycentre. Les coordonnées du barycentre donnent une bonne approximation de la valeur du décalage.

Le problème de cette première approximation est qu'elle se base uniquement sur le champ électrique à la surface du silicium. Or, on cherche avant tout à égaliser les puis-

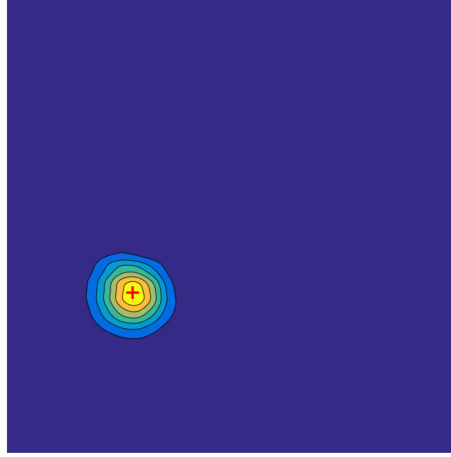


Figure 2.9. Exemple de barycentre de la tache de diffraction mesurée à la surface du silicium. La surface représentée en bleu fait la taille du quad-pixel soit $3.5\ \mu\text{m}$ de côté.

sances absorbées entre les sous-pixels. De plus, elle ne tient pas compte non plus des structures présentes à l'intérieur du silicium, typiquement les tranchées d'isolation, qui diffractent elles-aussi la lumière. Il est possible d'améliorer la précision en menant plusieurs autres simulations. En se basant sur le premier décalage trouvé par la méthode du barycentre, on peut simuler d'autres décalages autour, par exemple de $\pm 200\ \text{nm}$ sur les deux axes. Cela donne neuf simulations supplémentaires qui donnent la puissance absorbée par sous-pixel en fonction des décalages Δ_x et Δ_y . Si les neuf points de mesure sont assez proches, on peut faire deux hypothèses. D'une part, on peut considérer que la variation de flux en fonction des décalages est linéaire. D'autre part, la puissance reçue totale, c'est-à-dire la somme des puissances par sous-pixel, est constante quel que soit le décalage. Sous ces conditions, on peut faire passer un plan par ces neuf points de mesure, par sous-pixels. On a donc cinq plans au total, quatre pour les sous-pixels et un plan pour la puissance totale. L'idée est de calculer les lignes d'intersection entre les quatre plans des sous-pixels et le plan de la puissance totale. Il est aussi possible de calculer les intersections entre les quatre plans des sous-pixels entre eux. La Figure 2.10 montre ces différentes projections. Lorsque l'on projette ces lignes sur le plan $z = 0$ on observe qu'elles se croisent deux à deux, c'est donc que la puissance absorbée est égale entre deux sous-pixels (au moins). On observe tout de même un "encadrement" d'un point particulier, l'intersection des plans des sous-pixels en diagonale : le couple (A,D) en jaune et (B,C) en violet sur la figure. Ce point a la particularité de représenter l'égalité de la puissance absorbée par les quatre sous-pixels en même temps, ce sont donc les coordonnées de ce point que l'on considère être les valeurs optimales de décalage.

Le Tableau 2.2 regroupe les décalages calculés pour les différentes valeurs de CRA, en considérant le quad-pixel de $3.5\ \mu\text{m}$.

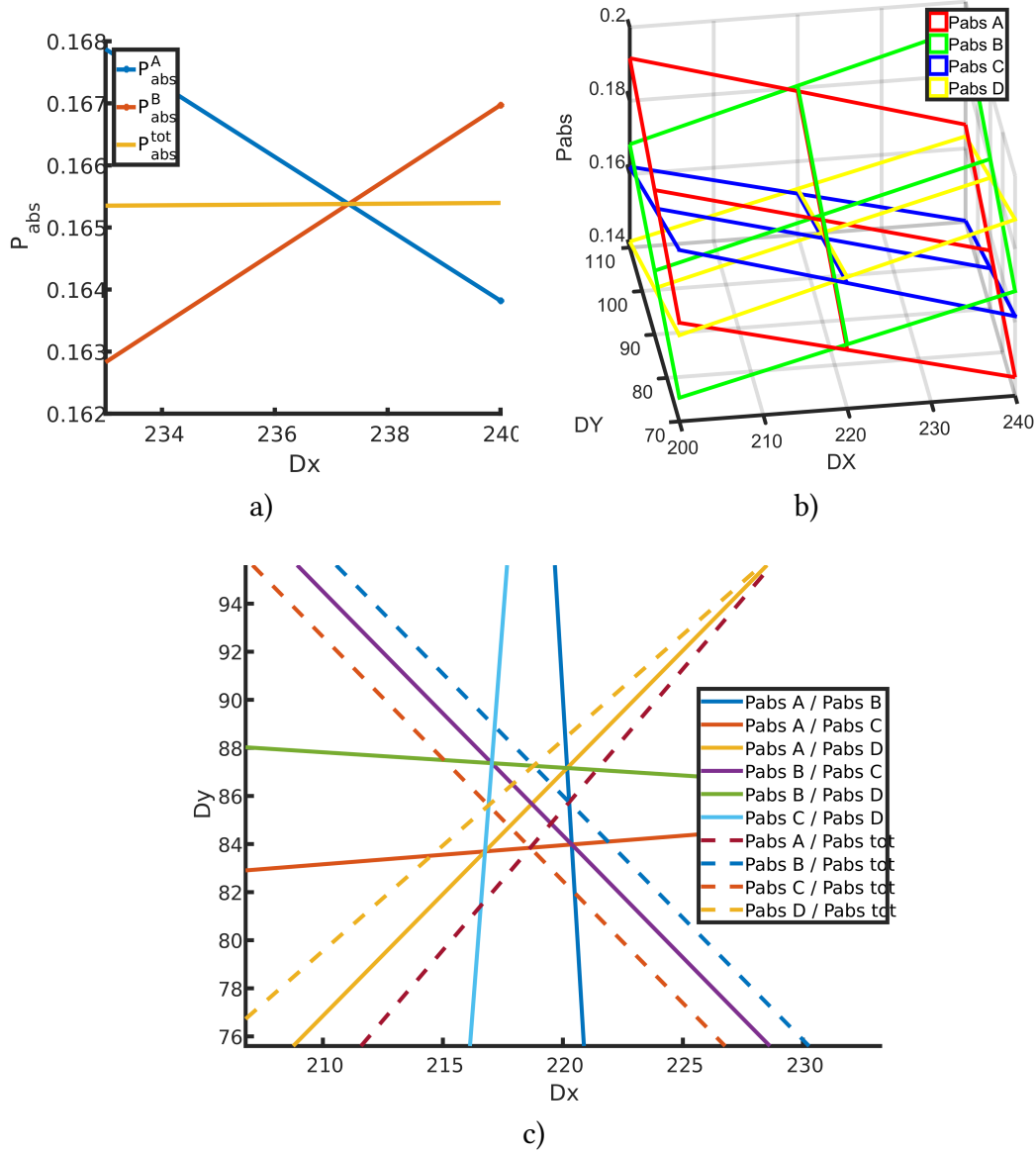


Figure 2.10. Illustration de la méthode pour trouver les décalages. La figure a) montre son application pour une seule direction. Les puissances absorbées par les parties gauche et droite du quad-pixel sont représentées en bleu et en orange. La puissance totale est en jaune. La coordonnée du point d'intersection des trois lignes donne la valeur du décalage. Pour un décalage dans les deux directions, on forme des plans avec neuf simulations. La figure b) montre les plans de chaque sous-pixel et la figure c) montre les lignes d'intersection entre ces plans projetées sur le plan $z = 0$. Les valeurs des décalages en x et y sont données par les coordonnées du point d'intersection central. L'unité des axes D_x et D_y est en nanomètres.

Zone	φ	θ	Δ_x	Δ_y
1	0	5	0.237	0
2	0.00	10.00	0.473	0.000
3	0.00	15.00	0.722	0.000
4	0.00	20.00	0.959	0.000
5	0.00	25.00	1.221	0.000
6	0.00	30.00	1.467	0.000

Zone	φ	θ	Δ_x	Δ_y
7	11.25	5.00	0.232	0.044
8	11.25	10.00	0.463	0.088
9	11.25	15.00	0.702	0.133
10	11.25	20.00	0.940	0.183
11	11.25	25.00	1.198	0.238
12	11.25	30.00	1.439	0.292

Zone	φ	θ	Δ_x	Δ_y
13	22.50	5.00	0.219	0.086
14	22.50	10.00	0.440	0.178
15	22.50	15.00	0.661	0.278
16	22.50	20.00	0.888	0.373
17	22.50	25.00	1.116	0.463
18	22.50	30.00	1.356	0.558

Zone	φ	θ	Δ_x	Δ_y
19	33.75	5.00	0.196	0.126
20	33.75	10.00	0.397	0.267
21	33.75	15.00	0.589	0.399
22	33.75	20.00	0.804	0.525
23	33.75	25.00	1.005	0.672
24	33.75	30.00	1.228	0.821

Zone	φ	θ	Δ_x	Δ_y
25	45.00	5.00	0.165	0.165
26	45.00	10.00	0.341	0.341
27	45.00	15.00	0.500	0.500
28	45.00	20.00	0.678	0.678
29	45.00	25.00	0.856	0.856
30	45.00	30.00	1.035	1.035

Tableau 2.2. Description des décalages CRA

2.2.3 Résultats en CRA

En se basant sur les décalages trouvés dans la partie précédente, j'ai simulé les réponses angulaires complètes pour certaines valeurs de CRA, toujours en considérant le quad-pixel $3.5\ \mu\text{m}$. Cela fournit des réponses angulaires en 4D ($\varphi_{cra}, \theta_{cra}, \varphi, \theta$) par sous-pixels. Pour éviter un trop grand nombre de simulations, voilà les angles qui ont été testés :

- φ_{cra} : $0^\circ, 22.5^\circ, 45^\circ$
- θ_{cra} : $0^\circ, 10^\circ, 20^\circ, 30^\circ$
- θ : de 0° à 40° par pas de 5° , puis de 40° à 70° par pas de 10°
- φ : de 0° à 360° par pas de 22.5°

Comme dit précédemment, il est possible de compléter la mesure grâce à la symétrie du quad-pixel, en permutant les réponses des sous-pixels selon l'angle d'incidence et selon le CRA. La Figure 2.11 regroupe toutes les réponses angulaires du sous-pixel A. Les parties entourées en rouge montrent les zones utiles de la réponse angulaire. Les parties non entourées représentent le crosstalk entre les microlentilles. Les parties utiles se déplacent avec le CRA de façon à ce que les puissances absorbées par les trois autres sous-pixels (non représentés) soient égales pour le CRA considéré.

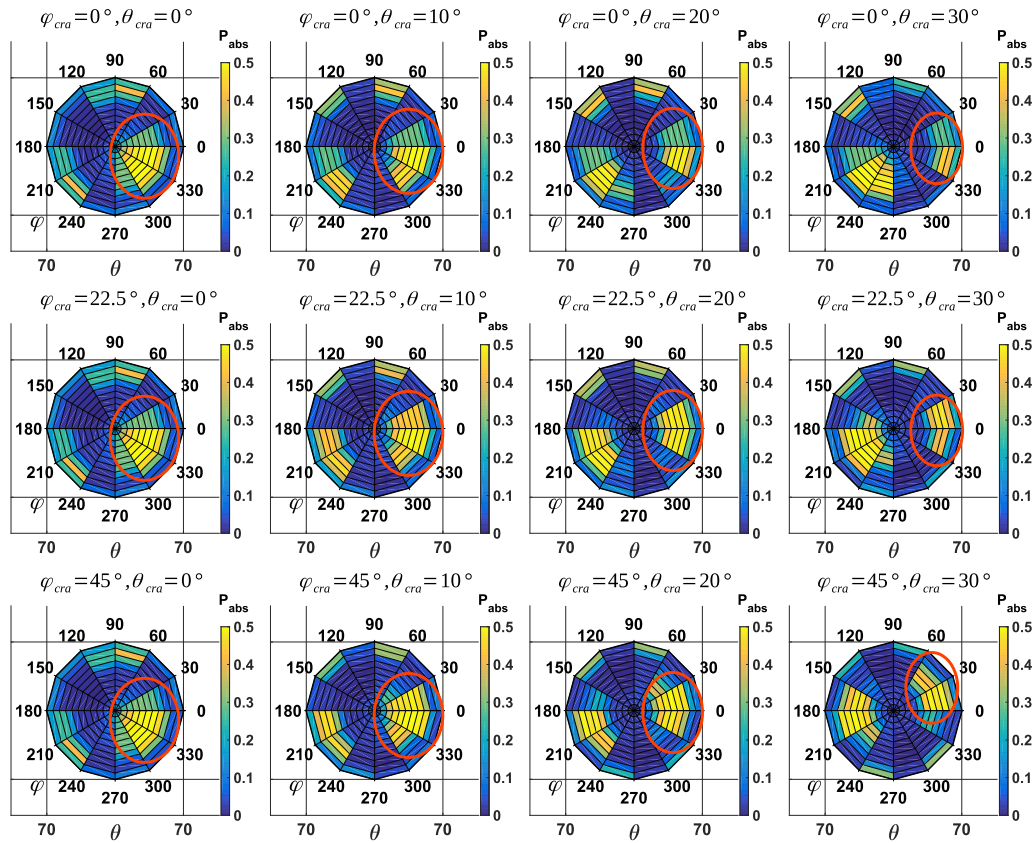


Figure 2.11. Réponses angulaires en 4D du sous-pixel A pour les différentes valeurs de CRA.

2.2.4 Analyse en longueur d'onde

Dans toutes les parties précédentes, j'ai travaillé avec une longueur d'onde verte. Dans cette partie, j'aborde brièvement la réponse en longueur d'onde. Je présente les réponses

angulaires et les ratios en 1 dimension pour les longueurs d'onde 400,532 et 700 nanomètres, respectivement simulées avec un filtre de couleur bleu, vert et rouge. J'ai simulé une microlentille centrée puis une microlentille décalée pour un CRA de $\varphi_{cra} = 0^\circ$, $\theta_{cra} = 25^\circ$. La longueur d'onde rouge est celle qui est la plus diffractée, et dont les performances sont les moins bonnes. La longueur d'onde bleue, au contraire, présente les meilleures performances. Dans le cas d'un quad-pixel destiné à l'imagerie visible, on place un motif de Bayer par microlentille donc au global sur le capteur, les performances moindres des microlentilles rouges seront compensées par les microlentilles bleues, et ça ne devrait pas gêner une mesure d'autofocus par exemple.

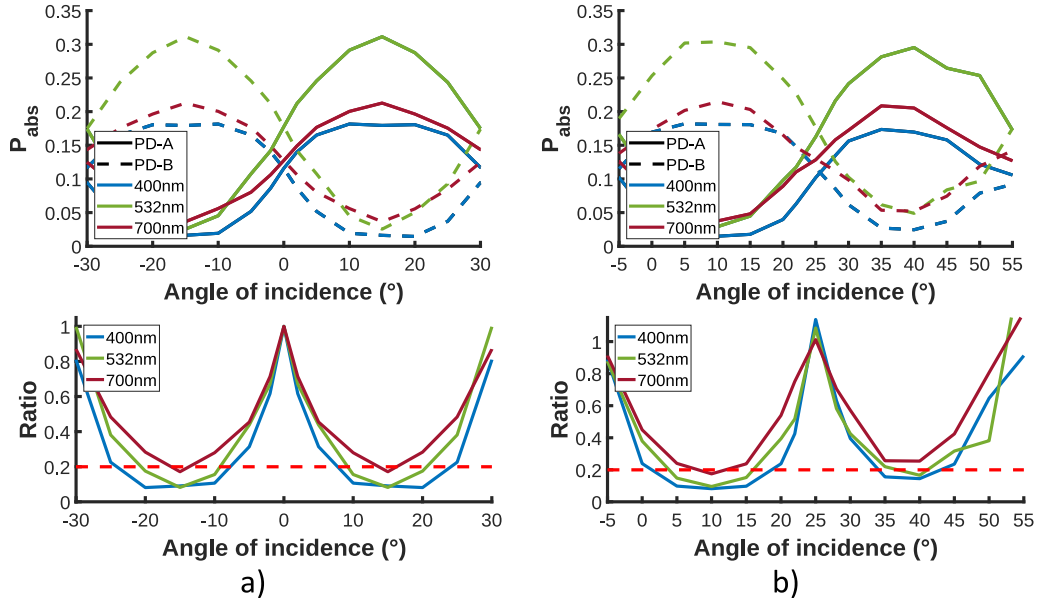


Figure 2.12. Réponses angulaires 1D pour différentes longueurs d'onde.

2.3 Comparaison avec l'état de l'art

2.3.1 Dual pixel

La technologie "Dual-Pixel" étant utilisée commercialement dans des caméras ou des téléphones depuis 2013 avec l'introduction du Canon 70D, différentes études ont été publiées depuis. Les deux principales contributions viennent de Canon (Kobayashi et al.[42]) et Samsung (Choi,Yun [19, 87]). Dans le premier cas, l'étude porte principalement sur des pixels d'appareils photos reflex ou sans miroir. Les microlentilles sont relativement grandes ($6.4\mu\text{m}$), centrées sur les sous-pixels, et ces derniers sont illuminés par la face avant (FSI, Front Side Illuminated) c'est-à-dire que l'on trouve des niveaux métalliques entre la microlentille et le silicium. L'auteur présente aussi les réponses angulaires pour une microlentille de $1.8\mu\text{m}$ en BSI. Dans le second cas, l'étude se concentre sur des pixels de smartphones BSI et dont les sous-pixels ont une taille inférieure au micron ($0.8\mu\text{m}$, $0.64\mu\text{m}$ voire $0.61\mu\text{m}$). L'article de Yun présente le cas d'une microlentille centrée, puis décentrée pour un CRA de 30° . Leurs résultats sont présentés respectivement par les figures Figure 2.13 et Figure 2.14.

Par rapport au pixel $6.4\mu\text{m}$ de Canon, notre pixel de $3.5\mu\text{m}$ obtient les mêmes performances malgré sa taille 1.8 fois plus petite. Concernant les sous-pixels dont la taille est inférieure au micron (Canon : $0.9\mu\text{m}$, Samsung : $0.61\mu\text{m}$), notre pixel de $1.6\mu\text{m}$ (sous-pixels de $0.8\mu\text{m}$) obtient des résultats comparables. On peut néanmoins noter que le pixel de Samsung, étant le plus petit présenté ici, n'arrive pas à atteindre un contraste angulaire de 5. Dans leur article les auteurs ont mesuré ce dernier à 3.5. Cette différence de contraste peut influencer sur les performances des algorithmes utilisés. J'étudie dans la partie suivante les plages angulaires R_a et R_b en fonction de la taille de la microlentille.

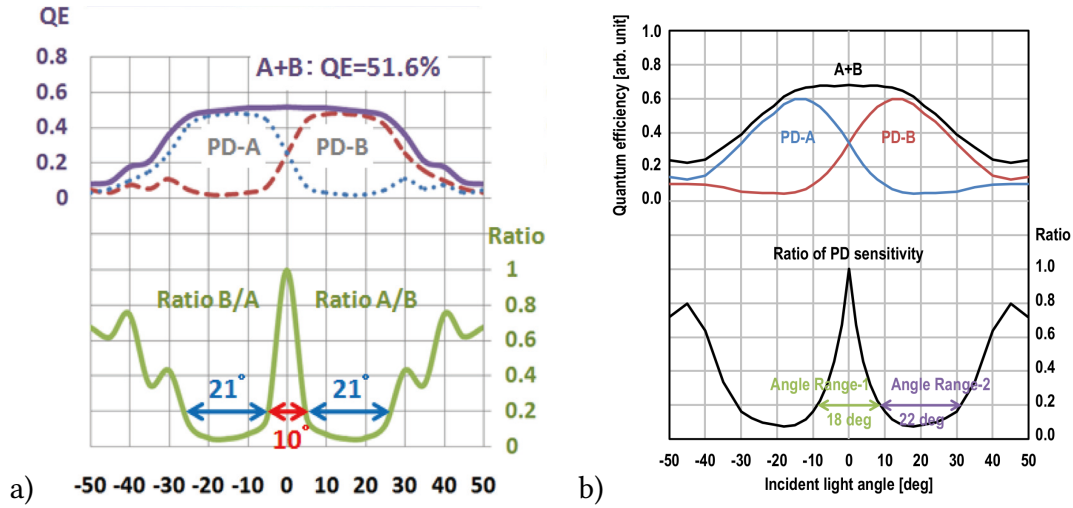


Figure 2.13. Réponses angulaires d'un dual-pixel Canon de $6.4\mu\text{m}$ en a) et $1.8\mu\text{m}$ en b). Illustration tirée de Kobayashi[42].

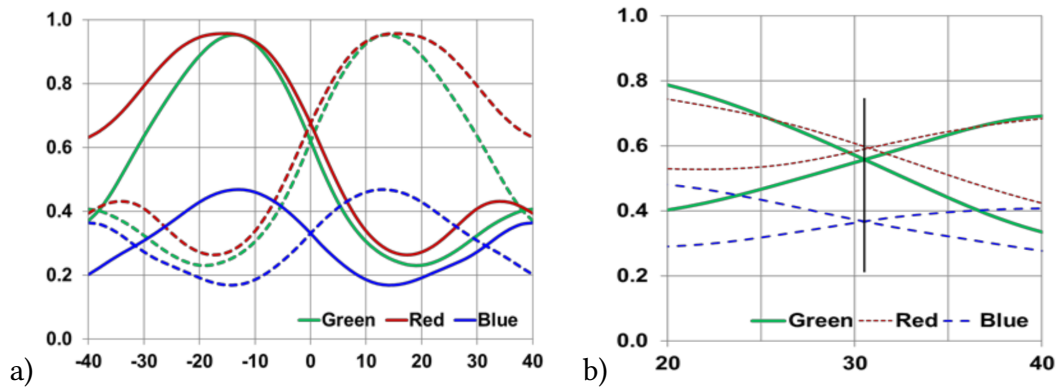


Figure 2.14. Réponses angulaires d'un dual-pixel Samsung de $0.61\mu\text{m}$, microlentille centrée en a) et décentrée en b). Illustration tirée de Yun[87]

2.3.2 Impact de la taille de la microlentille

Comme l'échantillonnage angulaire diminue la résolution spatiale, une manière de compenser est de diminuer la taille des microlentilles pour en placer un plus grand nombre

pour une taille de capteur donnée. La principale limitation vient de la diffraction qui cause du crosstalk entre les SAIs et dont l'effet augmente plus la microlentille est petite. Or on souhaite conserver un contraste angulaire élevé sur la plus grande plage possible. Le pire cas possible arrive lorsque les SAIs sont identiques indépendamment du défocus (et par extension de l'angle d'incidence), l'effet plénoptique est alors perdu. J'ai simulé plusieurs tailles de microlentilles, de $1.2\ \mu\text{m}$ à $6\ \mu\text{m}$. Leurs rayons de courbure ont été choisis le plus petit possible en fonction de la taille des sous-pixels. Leurs hauteurs ont été rapidement optimisées en simulant chaque microlentille et en mesurant le champ électrique dans le plan xy pour déterminer la hauteur où l'énergie était la plus concentrée. Les résultats sont montrés sur la Figure 2.15, où l'on trace la taille de la microlentille en fonction des plages angulaires R_a et R_b . Kobayashi a mené une étude similaire, sans doute en optimisant chaque dual-pixel de manière plus poussée. En effet, leurs résultats sont meilleurs que les nôtres. Par exemple, leur pixels de $1.2\ \mu\text{m}$ a un R_a de 20° et un R_b de 18° . Notre quad-pixel de $1.2\ \mu\text{m}$ a un R_a de 40° et un R_b de 0° (il n'atteint jamais un contraste angulaire de 5, à l'instar du dual pixel $0.61\ \mu\text{m}$ de Samsung).

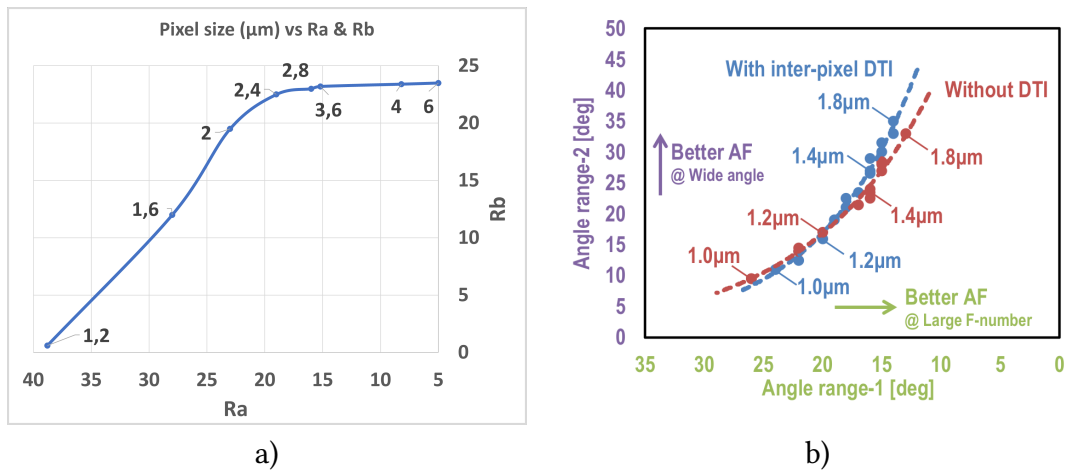


Figure 2.15. Impact de la taille de la microlentille sur les plages angulaires. Nos résultats sont à gauche, pour des tailles de microlentilles allant de $1.2\ \mu\text{m}$ à $6\ \mu\text{m}$. Ceux de Canon à droite, pour des tailles de microlentilles allant de $1.0\ \mu\text{m}$ à $1.8\ \mu\text{m}$. Les axes des abscisses et des ordonnées sont respectivement les plages angulaires R_a et R_b . La taille des microlentilles est notée le long de la courbe.

Le contraste angulaire est fixé selon l'application souhaitée, par exemple le contrôle de l'autofocus uniquement ou les capacités plénoptiques avancées, mais aussi selon la capacité des algorithmes à calculer les disparités entre les SAIs et la précision désirée. La valeur de contraste choisie permet de calculer la plage R_a que l'on veut en général la plus réduite possible, et la plage R_b que l'on veut la plus grande possible. Ces deux plages angulaires peuvent être transposées à la pupille de sortie de l'objectif principal utilisé avec le capteur. R_a correspond à une "zone morte" de l'objectif dans le sens où elle n'apporte pas d'information de parallaxe. Cette information est apportée par les quatre quarts de disque à la périphérie de l'objectif. R_b correspond à l'ouverture maximale de l'objectif utilisable avant de voir apparaître du crosstalk entre les microlentilles à cause d'un angle d'incidence trop élevé. Le schéma de la Figure 2.16 illustre ces différentes parties. Ce schéma correspond au ratio 2D présenté Figure 2.5, mais projeté sur la pupille de sortie de l'objec-

tif en fonction des angles d'incidence et de la focale. L'étendue de la zone morte dépend du contraste angulaire fixé : pour un même design de pixel, plus le contraste désiré est élevé plus la zone aura tendance à être grande, inversement plus on réduit la contrainte plus la zone sera réduite.

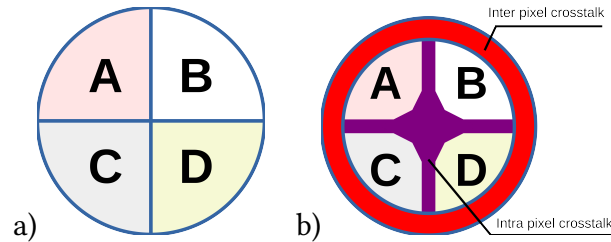


Figure 2.16. Illustration des parties utiles (A,B,C,D) et non utiles (violettes et rouges) de la pupille de l'objectif utilisé devant le capteur. Idéalement, on doit avoir un découpage net comme sur la figure a), mais la diffraction produit des "zones mortes" qui n'apportent pas d'information de parallaxe (en violet sur la figure b). De plus, si l'objectif est trop ouvert, cela peut causer du crosstalk entre les microlentilles (en rouge sur la figure b).

Par exemple, si la gestion de l'autofocus nécessite un contraste de cinq et si on se base sur notre quad-pixel de $3.5\ \mu\text{m}$ dont $R_a = 10^\circ$ et $R_b = 20^\circ$, il faut utiliser un objectif dont l'ouverture est plus grande que F/5.7 (angle d'incidence de $R_a/2 = 5^\circ$) pour pouvoir tirer un signal de parallaxe, mais plus petite que F/1.1 (angle d'incidence de $R_a/2 + R_b = 25^\circ$) pour éviter le crosstalk entre microlentilles. Cela convient pour les smartphones modernes dont les objectifs ont une ouverture comprise en F#1.2 et F#2.8 selon la gamme, cependant cela peut être un facteur limitant pour les appareils photographiques à objectifs interchangeables. On trouve par exemple des objectifs haut de gamme ouverts à F#0.95 qui ne permettront pas d'atteindre une bonne qualité d'image à cause du crosstalk s'ils sont utilisés à pleine ouverture. À l'inverse, il existe aussi des objectifs assez peu ouverts (par exemple Canon[14] 600 et 800mm F#11) dont l'autofocus sera moins véloce que d'autres téléobjectifs plus ouverts. Enfin, le problème se pose en vidéo où l'on veut pouvoir conserver un autofocus continu mais qui sera plus ou moins dégradé selon l'ouverture choisie.

2.4 Pistes d'améliorations des performances

Dans les parties précédentes, j'ai principalement abordé l'optimisation du point de vue de la microlentille : son rayon de courbure, sa hauteur et son positionnement. On peut jouer sur d'autres paramètres pour améliorer les performances. Comme on l'a vu auparavant, c'est surtout les petits pixels (globalement inférieurs à $2\ \mu\text{m}$) qui nécessitent une optimisation fine. L'effet de la diffraction est bien moindre sur des grands pixels et leurs performances sont déjà satisfaisantes. D'autre part, on cherche à réduire la taille des pixels pour gagner en résolution spatiale. Dans cette partie, je présente des designs de quad-pixels plus poussés qui visent à diminuer le crosstalk entre les microlentilles, puis entre les sous-pixels.

2.4.1 Diminution du crosstalk inter microlentilles

2.4.1.1 Barrière d'isolation de faible indice de réfraction

Il est possible de créer des barrières d'isolation entre les microlentilles dans l'épaisseur disponible entre le silicium et la base de la microlentille. Une barrière métallique est certainement meilleure en termes d'isolation pure, mais d'une part elle peut absorber et réfléchir la lumière incidente, ce qui diminue la puissance reçue, et d'autre part elle peut être difficile à appliquer dans le cadre de très petits pixels car la barrière aura une épaisseur minimale et masquera peut être une surface trop importante du silicium. Une meilleure solution consiste à placer des barrières d'isolation créées dans un matériau d'indice de réfraction plus faible que celui des autres couches. La Figure 2.17 montre un design de pixel avec des barrières d'indice $n = 1.3$ et de 100 nm de large sur un pixel de $1.6 \mu\text{m}$ (sous-pixels de $0.8 \mu\text{m}$). Ces matériaux à bas indice de réfraction existent et sont intégrables dans le processus de fabrication d'un capteur. Samsung utilise déjà ces matériaux pour leur technologie dite "Isocell +" [59]. Dans mes simulations, le gain sur la plage R_b s'élève à 3.7° .

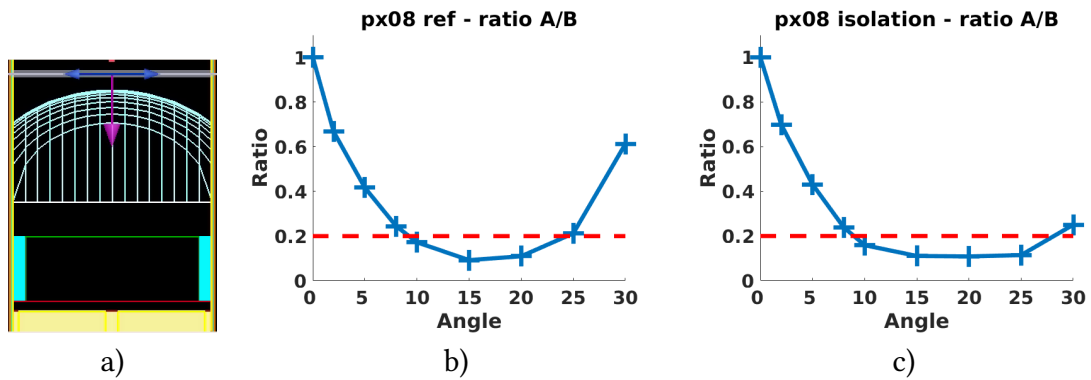


Figure 2.17. Schéma des isolations en matériau $n = 1.3$ sur la figure a). Elles sont visibles sur les côtés du filtre coloré en cyan. Les figures b) et c) représentent les ratios Gauche/Droite d'un pixel de $1.6 \mu\text{m}$, sans et avec les barrières respectivement. Le gain sur R_b est de 3.7°

2.4.1.2 Structure gravée dans le silicium

Une autre idée est de graver une structure spécifique dans le silicium, ce qui a pour but de diminuer le crosstalk entre les microlentilles. Cela a deux intérêts principaux : faire en sorte que le capteur accepte des objectifs plus ouverts et le rendre plus robuste face au CRA. L'idée de cette structure est de préserver la convergence de la microlentille à des angles d'incidence faibles, donc de garder R_a identique, mais de "casser" sa convergence à des angles d'incidence élevés et faire en sorte que le silicium absorbe l'énergie avant qu'elle ne passe dans la microlentille adjacente. Les détails de cette structure ne peuvent pas être divulgués pour l'instant.

Les figures Figure 2.18 et Figure 2.19 présentent les résultats en terme de ratio gauche/droite pour des sous-pixels de taille $1.75 \mu\text{m}$ et $0.8 \mu\text{m}$ respectivement. Le gain est d'au moins 10° pour le pixel de $3.5 \mu\text{m}$ (on sort des limites de la simulation), et de 5° pour le pixel de $1.6 \mu\text{m}$. La plage angulaire R_a est légèrement moins bonne que les performances

de référence. Cela est dû au fait que la structure n'a pas été finement optimisée et qu'elle dépend de la distance entre la microlentille et le silicium. Or, pour rester réaliste par rapport aux contraintes de fabrication, mais aussi conserver une bonne épaisseur pour le filtre coloré, cette distance n'a pas été réduite autant que possible dans les simulations.

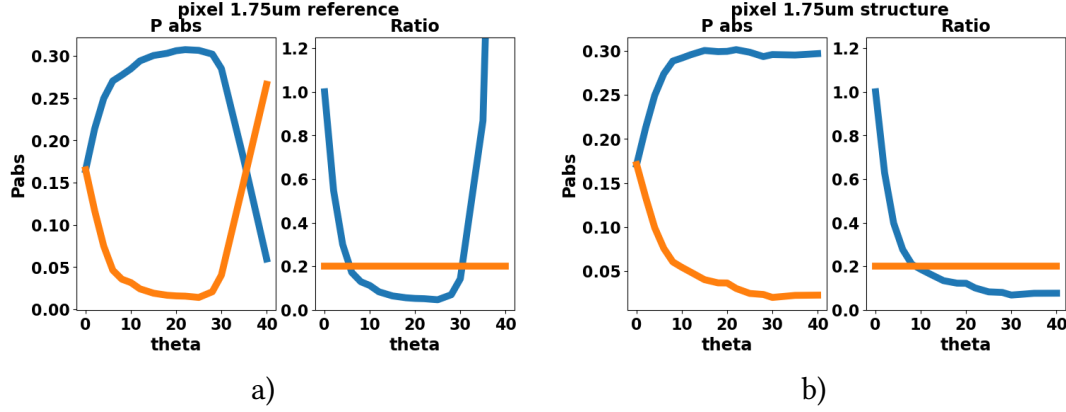


Figure 2.18. Résultat des simulations en sous-pixel 1.75 μm . La figure a) présente le pixel de référence, et la figure b) le pixel avec la structure. Le gain sur R_b est d'au moins 10°.

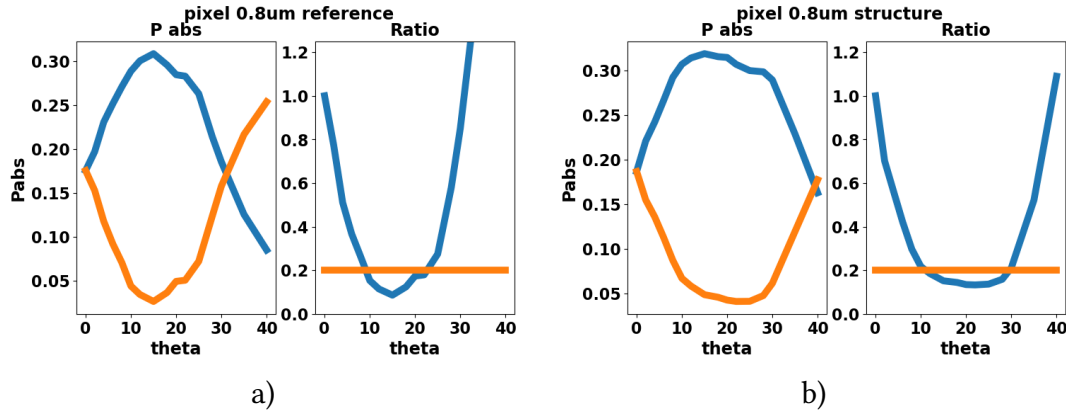


Figure 2.19. Résultat des simulations en sous-pixel 0.8 μm . La figure a) présente le pixel de référence, et la figure b) le pixel avec la structure. Le gain sur R_b est d'environ 5°.

J'ai basé la structure sur un procédé de gravure standard du silicium. Il serait possible d'utiliser un procédé plus avancé, par exemple la lithographie en niveau de gris[18], pour ajuster la forme plus finement. D'une manière générale, la structure offre des degrés de liberté supplémentaires lors de l'optimisation du pixel. De plus, il est possible de généraliser cette solution pour du dual-pixel ou du nona-pixel.

2.4.2 Diminution du crosstalk intra microlentilles

Réduire le crosstalk entre les sous-pixels est plus complexe que diminuer celui entre les microlentilles car il est directement causé par la diffraction de la microlentille. Nous avons essayé de simuler d'autres designs de quad-pixel, par exemple une seconde couche

de microlentilles en nitrure de silicium, soit globale soit par sous-pixel (Figure 2.20 a) et b). Nous avons aussi essayé de placer des barrières d'isolation entre les filtres colorés de chaque sous-pixel (Figure 2.20 c).

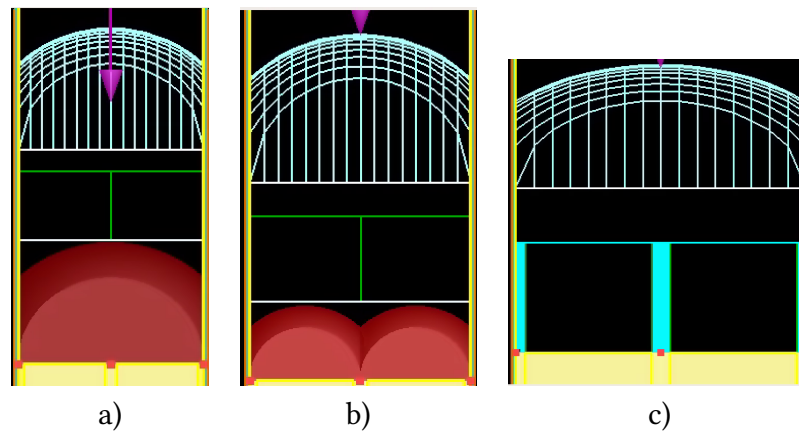


Figure 2.20. Schémas de designs alternatifs de quad-pixels $1.6\ \mu\text{m}$: a) une seconde microlentille, b) une seconde microlentille par sous-pixel, c) une isolation bas indice dans le filtre coloré. **Note** : la microlentille en nitrure de silicium est en-dessous des filtres colorés car on fabrique d'abord les couches inorganiques (silicium) puis les couches organiques (résines, microlentilles).

Le meilleur résultat en termes de gains sur la plage R_a est de placer des barrières de bas indice de réfraction en forme de croix dans le filtre coloré. On gagne alors 2° sur la plage R_a . Cependant, on perd beaucoup sur la plage R_b , environ 5° , comme on peut le voir sur la Figure 2.21.

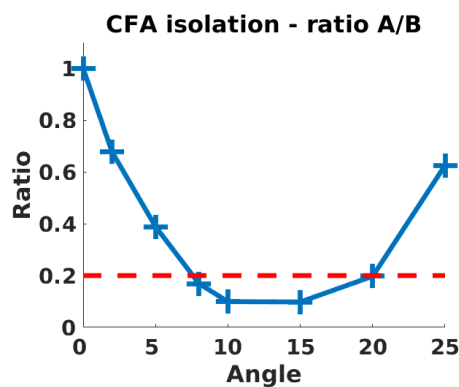


Figure 2.21. Ratio gauche/droite du pixel avec des isolations dans la résine colorée, fabriquées avec un matériau de bas indice de réfraction.

Les procédés de fabrication et l'utilisation de microlentilles réfractives imposent une certaine distance entre le silicium et la microlentilles et cela devient limitant pour des petits pixels. Des technologies planaires diffractives comme les Fresnel Zone Plate, ou les méta-surfaces pourraient aider à améliorer les performances dans ce cas. Cependant, leur utilisation semble limitée au premier abord. En effet, il faut que l'élément qui remplace la

microlentille réfractive fonctionne pour une large gamme d'angles d'incidence et soit en même temps peu dépendant de la longueur d'onde, ce qui n'est pas le cas actuellement. Enfin, il faut que ce nouvel élément puisse être fabriqué dans des tailles égales ou inférieures à $4\text{ }\mu\text{m}$ environ pour être utiles car les performances des microlentilles réfractives sont très correctes pour de grand pixels.

2.5 Conclusion

Dans ce chapitre, j'ai présenté mes travaux en simulations électromagnétiques d'un quad-pixel. J'ai explicité les critères de performance utilisés pour l'optimisation des différents éléments du pixel, les méthodes de placement des microlentilles lorsque l'angle du rayon principal est non nul, et enfin j'ai montré les réponses angulaires complètes, en 2D pour un quadpixel de $1.6\text{ }\mu\text{m}$ et en 4D pour le quadpixel de $3.5\text{ }\mu\text{m}$. Ces simulations sont importantes pour mieux comprendre la formation des images par un capteur plénoptique, mais elles se sont avérées aussi utiles pour deux autres aspects de ma thèse. Le premier point concerne le prototype réalisé au CEA. L'optimisation du pixel et les calculs des décalages du CRA permettent de définir un cahier des charges précis. Ensuite, je montre dans le chapitre suivant que l'on peut intégrer les résultats dans un logiciel de rendu par tracé de rayons, et ainsi générer des images prenant en compte la diffraction. Ces images de synthèse m'ont permis de travailler sur les aspects de traitement d'images en avance de phase par rapport au prototype.

Une partie de ce travail a été publiée dans le journal Optic Express[16] et les structures développées pour réduire le crosstalk entre les microlentilles ont fait l'objet d'un brevet déposé conjointement par Interdigital et le CEA.

Rendu en lancé de rayons

Dans ce chapitre, je présente une modélisation de la formation des images de sous-ouverture (SAI) qui prend en compte à la fois l'objectif utilisé pour former l'image de la scène et les effets de la diffraction au sein du pixel. À partir de cette modélisation, j'ai modifié un logiciel de rendu d'images par lancé de rayons pour qu'il prenne en compte les réponses angulaires des pixels. Cela permet de produire des images plus réalistes du point de vue du capteur et permet d'étudier les effets de la diffraction sur les SAIs. De plus, ces images permettent de travailler sur les algorithmes de traitement d'images sans pour autant posséder un capteur plénoptique quad-pixel. En particulier, cet outil nous a permis de générer un dataset d'images quad-pixel et ainsi de travailler sur des techniques d'apprentissage profond.

Dans une première partie, je décris un modèle de la fonction de flou associée à chaque SAI basé sur la réponse angulaire. J'explique ensuite les modifications apportées à notre logiciel de rendu pour prendre en compte la diffraction dans le pixel et je décris comment j'ai validé la méthode. Enfin, je présente les images obtenues et étudie l'impact de la diffraction en prenant comme exemple la correction des aberrations de l'objectif principal.

3.1 Différentes approches pour le rendu d'images

On peut considérer deux approches pour générer des images issues d'un capteur plénoptique. La première est d'utiliser des images nettes, déjà existantes, puis de les convoluer avec une fonction de flou qui correspond à l'effet qu'on veut simuler. Je présente deux modèles dans la partie suivante, l'un issu d'un article de 2020 sur le dual pixel, l'autre basé sur nos connaissances des réponses angulaires. L'autre approche est le rendu en lancé de rayons, où une image est générée à partir d'une scène 3D construite au préalable.

L'avantage de la première approche est qu'une convolution est relativement rapide à calculer, au plus quelques minutes selon la taille de l'image d'entrée et la taille du support de la fonction de flou. Son principal défaut est que la même fonction de flou est appliquée à toute l'image. Par exemple, si on veut simuler un flou de défocalisation lié à la profondeur, alors l'image résultante aura une profondeur constante. On pourrait éventuellement modifier cette méthode pour générer des images avec des zones à différentes profondeurs. Il faudrait segmenter l'image de départ selon la distance, puis générer une multitude de kernels différents en fonction de la profondeur, et enfin convoluer chaque zone avec le bon kernel. Bien que techniquement possible, cette méthode ne semble pas efficace, surtout si

le nombre de zones à différentes profondeurs augmente.

L'approche du rendu en lancé de rayons peut se décliner en trois cas :

- Si les pixels sont grands (supérieurs à une quinzaine de microns), on peut s'en tenir à l'optique géométrique et simuler des microlentilles idéales (lentilles minces paraxiales achromatiques). Le crosstalk entre les sous-pixels est totalement négligé.
- Si les pixels sont de taille moyenne (comprise entre $15\ \mu\text{m}$ et $5\ \mu\text{m}$), un compromis est de toujours utiliser l'optique géométrique, mais de simuler la vraie architecture des pixels utilisés (microlentille, résine colorée, passivation, etc). C'était d'ailleurs la méthode de simulation des pixels utilisée avant que les méthodes FDTD apparaissent. Le crosstalk entre les sous-pixels est approximé par les aberrations géométriques du vrai empilement.
- Si les pixels sont petits (inférieurs à $5\ \mu\text{m}$), l'approximation de l'optique géométrique n'est plus valide et on doit forcément prendre en compte la diffraction par les microlentilles de manière précise. Je présente une méthode pour lier les réponses angulaires à la méthode de lancé de rayon dans les parties suivantes.

Le choix de la méthode dépend du type d'images à générer. Si la profondeur est constante on peut utiliser une convolution. Si l'on cherche à simuler une image complexe dont les objets sont à des profondeurs différentes, il vaut mieux utiliser le rendu par lancé de rayons.

3.2 Modélisation de la fonction de flou

3.2.1 Spot diagramme et champ de lumière

Un capteur plénoptique doit être utilisé avec un objectif pour produire l'image. Selon le niveau de défocus, les images d'un objet sur les quatre images de sous-ouverture (Sub-Aperture Images, SAI) peuvent apparaître translatées les unes par rapport aux autres et c'est ce décalage qui permet d'utiliser les algorithmes dédiés aux champs de lumière. On peut en fait considérer qu'un capteur plénoptique de type 1 (Section 1.4) échantillonne le champ de lumière issu de l'objectif utilisé. On définit habituellement le spot diagramme comme l'ensemble des impacts des rayons lumineux issus d'une source ponctuelle et passant à travers le système optique. Il est défini pour une distance à l'objet D et pour une position sur le capteur (x_s, y_s) ou un CRA $(\theta_{cra}, \varphi_{cra})$. Je propose ici d'étendre la notion de spot diagramme et d'y ajouter l'angle d'incidence de chaque rayon par rapport au plan image. Ce spot diagramme S est donc un champ de lumière réduit qui correspond à la paramétrisation (x, y, θ, φ) de chaque rayon issu d'une source ponctuelle placée à une distance D , passant par l'objectif et frappant le plan image autour des coordonnées (x_s, y_s) . Pour un système optique idéal tous les rayons frappent exactement le plan image aux coordonnées (x_s, y_s) , mais ce n'est pas le cas en général avec un système optique réel qui peut souffrir d'aberrations géométriques. S est un ensemble de rayons

$$S(D, x_s, y_s) = \{R_1, ..., R_i, R_N\} \quad (3.1)$$

où les rayons ont quatre coordonnées, leurs positions sur le capteur et leurs angles d'incidence

$$R_i = R_i(x_i^r, y_i^r, \varphi_i, \theta_i) \quad (3.2)$$

(x_s, y_s) and $(\theta_{cra}, \varphi_{cra})$ sont liés par la formule optique de l'objectif f_{cra} :

$$\theta_{cra} = CRA_{func} \left(\sqrt{x_s^2 + y_s^2} \right) \quad (3.3)$$

$$\varphi_{cra} = \text{atan2}(x_s, y_s) \quad (3.4)$$

CRA_{func} peut être calculé en utilisant un logiciel de conception optique comme Zemax ou Code V. Un exemple a été donné dans le chapitre précédent à la Figure 2.8 qui montrait la fonction CRA_{func} d'une lentille mince et d'un objectif décrit dans le brevet S8320061-B2.

Le spot diagramme étendu permet de décrire plus facilement la fonction de flou dans la partie suivante. Il faut noter que la paramétrisation du champ de lumière se fait de manière plus usuelle à l'aide de deux plans, en travaillant en coordonnées (x, y, u, v) où (x, y) sont les coordonnées dans le plan des microlentilles et (u, v) les coordonnées dans la pupille de sortie de l'objectif. Le choix de la paramétrisation (x, y, θ, φ) vient des simulations des pixels dans le chapitre précédent où l'on étudiait leurs réponses en angle. On peut ainsi directement utiliser les réponses angulaires données en (θ, φ) , cela évite une conversion de type $\theta = \arctan(|u - x|/f)$, où f est la focale de l'objectif.

3.2.2 Modélisation de la fonction de flou par image de sous-ouverture

Une modélisation de la fonction de flou a été proposée par Punnappurath[62] dans le cas du dual-pixel. Elle est valable pour une scène dont la profondeur est constante et pour des réponses angulaires qui décroîtraient de manière linéaire. Seule la partie de droite est calculée, celle de gauche étant obtenue par symétrie. Un disque est translaté de 0 à $|2s|$, s désignant le rayon du disque, puis multiplié terme à terme à un autre disque qui reste fixe au centre. A chaque pas en translation, le produit est ajouté au résultat global, puis ce dernier est normalisé à 1. L'équation du modèle est la suivante :

$$H_{r_s} = \sum_{i=0}^{|2s|} \left\{ C(0, 0) \odot C\left(\frac{s}{|s|}i, 0\right) \right\} \quad (3.5)$$

où H_{r_s} est la fonction de flou pour la SAI de droite, \odot la multiplication terme à terme, $|\cdot|$ la valeur absolue, et $C(x, y)$ le disque de rayon s et centré en (x, y) . On peut aussi voir ce modèle comme la fonction d'auto-corrélation d'un disque de diamètre s mais "cumulée dans l'espace". Un exemple du kernel produit est montré sur la Figure 3.1.

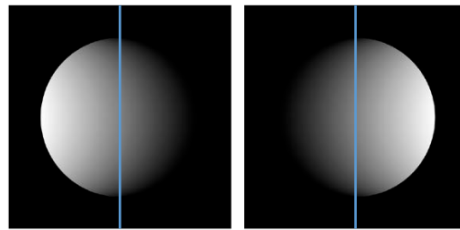


Figure 3.1. Exemple de kernel produit par le modèle de Punnappurath. Illustration tirée de leur article[62]

A partir de la connaissance des réponses angulaires du pixel, il est possible de décrire plus précisément la fonction de flou de chaque SAI en liant le spot diagramme de l'objectif

aux réponses angulaires du pixel. On considère une scène F à une distance constante D formant une image centrée sur les coordonnées (x_s, y_s) dans le plan image. Dans le cadre de l'imagerie incohérente, l'image finale est la scène convoluée par la fonction d'étalement du point (Point Spread Function, [PSF](#)). Il en découle alors pour un quad-pixel :

$$G^{sai}(D, x_s, y_s) = F * H^{sai}(D, x_s, y_s) \quad (3.6)$$

et

$$G = \sum_{sai} G^{sai} = F * \sum_{sai} H^{sai} = F * H \quad (3.7)$$

où G^{sai} est l'une des SAIs ($sai = \{A, B, C, D\}$), H^{sai} est la fonction de flou associée avec l'une de ces SAIs, G est l'image finale et H la fonction de flou globale. Cette formulation s'étend à toutes les tailles de capteur plénoptique en faisant varier le nombre de SAIs.

Je propose de décrire H^{sai} de manière plus précise. En première approximation, on peut négliger la diffraction produite par l'objectif principal et ainsi conserver une approche géométrique et utiliser la notion de spot diagramme / champ de lumière définie auparavant. Cette approximation peut se justifier car en général des objectifs très ouverts sont utilisés avec des capteurs plénoptiques. Comme on l'a vu au chapitre précédent, il faut que l'objectif produise des angles d'incidence plus élevés que la plage angulaire R_a pour produire un signal de disparités exploitable. L'angle des rayons est donc principalement affecté par le système optique et ses aberrations géométriques plutôt que par sa diffraction. Étant donné une réponse angulaire par SAI, normalisée à 1 :

$$A = A^{sai}(\theta, \varphi, x_s, y_s) \quad (3.8)$$

on peut écrire le kernel de flou par :

$$H^{sai}(D, x_s, y_s) = S(D, x_s, y_s) \otimes A^{sai}(\theta, \varphi, x_s, y_s) \quad (3.9)$$

\otimes correspond à une "multiplication angulaire". Cette opération pondère chaque rayon du spot diagramme par la valeur de la réponse angulaire en fonction de l'angle d'incidence et l'ajoute au kernel. $S \otimes A$ est une carte de poids

$$S \otimes A = \{w_1, \dots, w_i, w_N\} \quad (3.10)$$

où les poids ont pour coordonnées (x_i^w, y_i^w) et sont calculés comme suit :

$$w_i = w_i(x_i^w, y_i^w) = \sum_{\substack{r_j \in S \\ x_j^r = x_i^w, y_j^r = y_i^w}} r_j(x_j^r, y_j^r, \theta_j, \varphi_j) \times A(\theta_j, \varphi_j, x_s, y_s) \quad (3.11)$$

L'Équation (3.11) définit la multiplication angulaire pour le plan image en coordonnées réelles, où (x_i^w, y_i^w) sont les coordonnées d'intersection entre les rayons et le plan image. Il en allait de même pour les coordonnées spatiales du spot diagramme (x_s, y_s) et celles des rayons (x^r, y^r) . Pour prendre en compte l'échantillonnage spatial du capteur il faut additionner les rayons tombant sur les mêmes microlentilles. Dans ce cas-là (x_i^w, y_i^w) et (x_s, y_s) sont des coordonnées entières et représentent les indices des microlentilles sur le

capteur. Les coordonnées des rayons restent décimales. Soit m la taille d'une microlentille, on a :

$$w_i = w_i(x_i^w, y_i^w) = \sum_{\substack{r_j \in S \\ x_i^w - m/2 \leq x_j^r \leq x_i^w + m/2 \\ y_i^w - m/2 \leq y_j^r \leq y_i^w + m/2}} r_j(x_j^r, y_j^r, \theta_j, \varphi_j) \times A(\theta_j, \varphi_j, x_s, y_s) \quad (3.12)$$

Finalement, le kernel de flou associé à une SAI et une distance D est donné par :

$$G^{sai} = F * H^{sai} = F * (S \otimes A^{sai}) \quad (3.13)$$

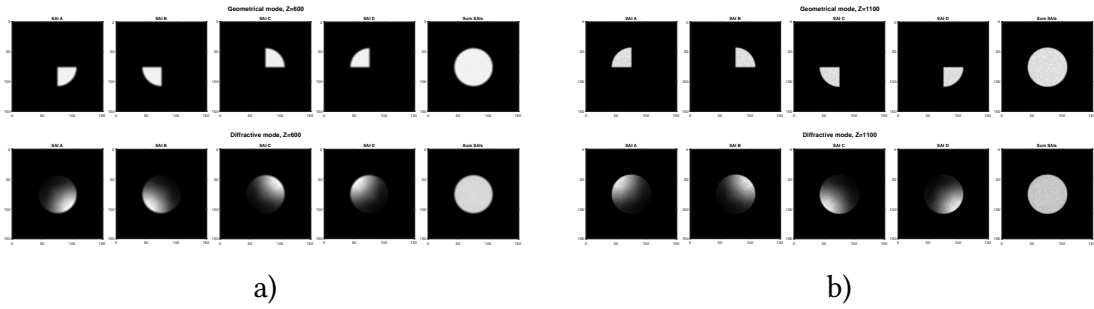


Figure 3.2. Kernels de flou simulés, avec $D_{focus} = 750mm$. La première et la seconde ligne de la figure a) montrent l'image d'une source ponctuelle située à $z = 600mm$, en mode classique et diffractif respectivement. b) montre la même chose mais pour une source située à $z = 1100mm$.

Les Figure 3.2 a) et b) montrent respectivement les SAIs d'une source ponctuelle en défocus, devant et derrière le plan de netteté. La première ligne représente le kernel de flou sans diffraction, et la seconde ligne avec la diffraction. Dans le premier cas, chaque SAI voit un quart de pupille parfaitement délimité, mais dans le second cas, le crosstalk entre les sous-pixels devient visible et les quarts de disques ne sont plus aussi bien découpés. Le modèle de Punnapurath fournit uniquement des disques dont l'intensité varie linéairement de gauche à droite, mais il reste indépendant des aberrations de l'optique utilisée et de l'architecture des pixels. Notre formulation permet d'inclure ces deux aspects du système d'imagerie.

3.3 Simulations croisées et rendu d'images

3.3.1 Présentation de l'outil de rendu existant

La technique de rendu en lancé de rayon (ray-tracing) permet de générer des images réalistes du point de vue de la propagation de la lumière dans l'espace et lors de la rencontre avec des objets opaques ou transparents. Le principe, ici simplifié, est de lancer des rayons depuis une position aléatoire au sein de chaque pixel du capteur, puis de les faire se propager à travers la scène en utilisant des équations de la physique (par exemple Snell-Descartes lors de la traversée d'un objet transparent). Les rayons peuvent être arrêtés selon plusieurs conditions, par exemple lorsqu'ils atteignent une source lumineuse,

lorsque la distance maximale de propagation sans intersection a été atteinte ou lorsqu'ils rencontrent un matériau opaque. Les rayons reviennent ensuite avec l'information d'intensité et de couleur sous la forme d'un triplet (R,G,B) et sont ajoutés au pixel d'où ils sont partis. Plusieurs rayons par pixel sont nécessaires pour diminuer le bruit de lancé de rayon.

Une version de PBRT v2 ([60, 61]) a été modifiée par l'équipe avec laquelle je travaillais pour simuler un capteur plénoptique. Je me suis basé sur cette version pour développer la méthode permettant de simuler les effets de la diffraction. Elle permet de placer une grille de microlentilles devant un groupe de pixels pour former un capteur plénoptique. Elle offre aussi la possibilité de simuler des surfaces sphériques ou asphériques, ce qui permet de créer et d'utiliser un objectif photographique complexe au lieu du sténopé habituel. Enfin, les formules de Cauchy et de Sellmeier ont été implémentées, ce qui permet de prendre en compte des effets de chromatisme. Cependant, comme tous les outils de rendu existants, elle se base sur une approche purement géométrique, même au niveau des pixels. Une microlentille idéale ne peut que produire des réponses angulaires parfaites, comme illustré dans les Figure 3.4 a) et Figure 3.4 b). De plus l'angle d'incidence maximal θ_L vaut :

$$\theta_L = \arctan\left(\frac{\text{taille sous-pixel}}{\text{focale de la microlentille}}\right) \quad (3.14)$$

La plage angulaire R_a vaut toujours 0 et le crosstalk entre les sous-pixels n'est pas pris en compte. Cela est dû au fait qu'en optique géométrique et dans le cadre d'un capteur plénoptique de type 1, un rayon partant d'un sous-pixel ne pourra jamais avoir un angle d'incidence qui n'est pas compris dans le quart de cône que le sous-pixel est censé couvrir. Plus simplement, dans le cas d'un dual-pixel par exemple, le sous-pixel de droite ne peut recevoir des rayons venant de la droite comme illustré sur la Figure 3.3.

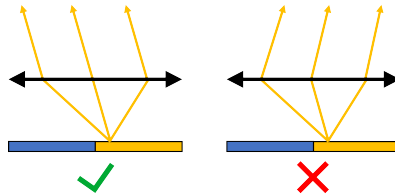


Figure 3.3. *Problème de l'optique géométrique pour la simulation de la diffraction. Une microlentille idéale placée à une distance au capteur égale à sa distance focale (configuration d'un capteur plénoptique de type 1) ne permet pas aux sous-pixels de recevoir des rayons venant de toutes les directions. Les rayons ne respectant pas la relation de conjugaison sont stoppés.*

3.3.2 Méthode d'intégration des réponses angulaires

La méthode consiste à ajouter l'information d'intensité et de couleur d'un rayon non seulement au sous-pixel d'où il est parti, mais aussi aux autres sous-pixels adjacents placés sous la même microlentille (Figure 3.4c). En procédant ainsi, chaque sous-pixel est capable de voir virtuellement toute la microlentille et pas seulement un quart. Avant d'ajouter le rayon à un sous-pixel, on le pondère par la valeur de la réponse angulaire (Figure 3.4 d) correspondante. L'algorithme détaillé étape par étape est le suivant :

- Un rayon est lancé depuis un sous-pixel jusqu'à la microlentille (celui de droite dans la Figure 3.4 c).
- L'angle d'incidence (φ, θ) est calculé après le passage par la microlentille. Si le CRA est corrigé on calcule aussi $(\varphi_{cra}, \theta_{cra})$.
- On laisse le rayon se propager à travers la scène, produisant un triplet (R,G,B).
- On lit la réponse angulaire et on calcule les puissances absorbées P_{abs} pour les 4 sous-pixels en fonction de l'angle d'incidence. On utilise une interpolation linéaire 2D (4D si le CRA est corrigé).
- Les valeurs (R,G,B) sont pondérées par P_{abs} (Figure 3.4 d) selon le sous-pixel puis ajoutées.

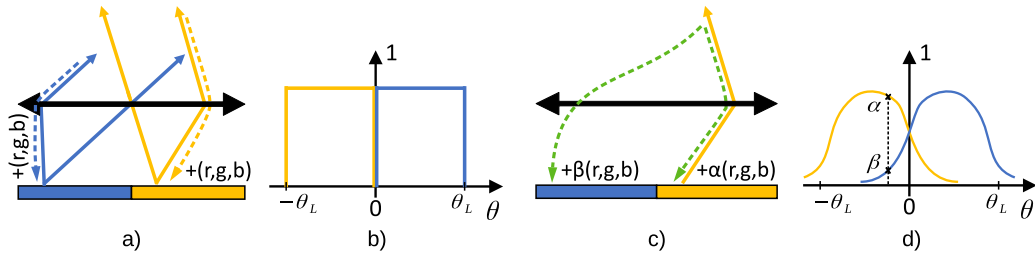


Figure 3.4. a) et b) montrent la procédure de lancé de rayon classique et la réponse angulaire idéale produite. c) et d) montrent notre méthode et la réponse angulaire réelle.

On continue d'utiliser des microlentilles idéales pour cette méthode. En effet, si l'on simulait le vrai design du pixel, il y aurait un conflit entre les effets de la diffraction que l'on simule via notre méthode et les aberrations géométriques de la surface sphérique de la microlentille. Il faut aussi adapter l'ouverture de la microlentille à l'ouverture de l'objectif utilisé : on veut que les rayons passent tous par la pupille de l'objectif principal et faire en sorte qu'ils parviennent tous jusqu'à la scène. En effet, le logiciel fait que si un rayon ne passe pas par la pupille de sortie de l'objectif, il est arrêté et donc cela augmente le bruit de lancé de rayon.

Bien que notre méthode d'intégration des réponses angulaires diffère légèrement de la formulation de la PSF présentée dans la partie précédente, des similarités sont présentes. On retrouve en particulier le fait de pondérer l'intensité du rayon en fonction de l'angle d'incidence. Cette méthode n'est pas limitée au quad-pixel et peut être facilement étendue à toutes les tailles p de capteur plénoptique. Il suffit de pondérer et d'ajouter chaque rayon partant d'un sous-pixel aux $p^2 - 1$ autres sous-pixels. Comme une seule simulation électromagnétique pour un angle d'incidence donné fournit les puissances absorbées par tous les sous-pixels, le nombre de simulations requises n'augmente pas. En outre, plus la taille de la microlentille diminue plus les effets de la diffraction s'accroissent et doivent être pris en compte. D'un point de vue des simulations FDTD, cela est assez rentable car le temps de simulation est proportionnel à la surface de la microlentille. On considère en effet que la hauteur du pixel varie peu, on conserve environ $4\mu\text{m}$ de silicium et au moins 500 nm de résine colorée quelle que soit la taille des microlentilles. Par contre, une limitation potentielle de cette méthode est le temps de lecture des réponses angulaires et leur interpolation : il faut procéder à p^2 interpolations quadri-linéaires lors du lancé de rayons.

3.3.3 Validation

Dans cette partie, je présente deux expériences qui visent à valider la méthode. Le but est de reproduire la réponse angulaire des simulations **FDTD** à l'aide d'un rendu **PBRT**. On considère un capteur constitué d'une seule microlentille et de 4 sous-pixels de $1.75\ \mu\text{m}$. Dans **PBRT**, la source lumineuse et la caméra ne peuvent pas se voir directement donc il faut forcément un objet qui réfléchit la lumière. On place donc la source derrière la caméra et une texture blanche lambertienne devant. Dans la première expérience, on place un disque de 5 mm à 1 m du pixel, cela doit simuler une source ponctuelle. En déplaçant ce disque sur les axes x et y , on simule plusieurs angles d'incidence. On doit ensuite passer des coordonnées (x,y) aux coordonnées (φ, θ) . Pour ce faire, on utilise une interpolation linéaire en mode backward.

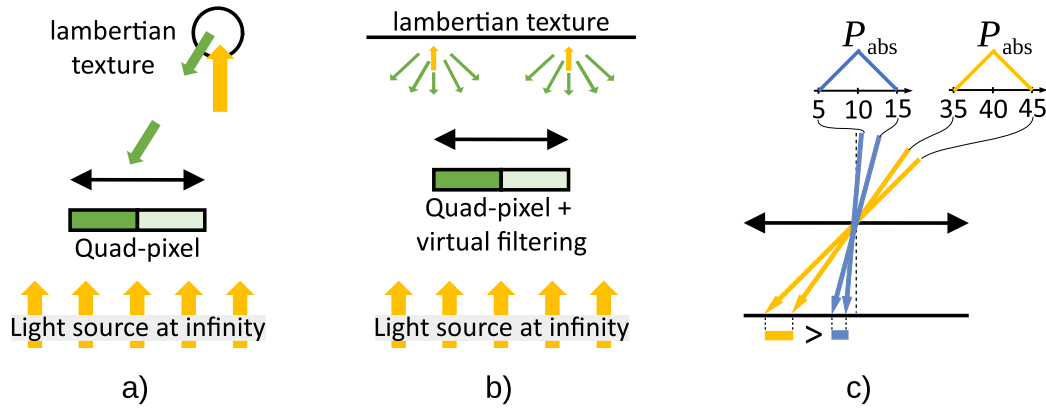


Figure 3.5. Description des expériences. La première (a) simule un disque qui se déplace dans le champ. La seconde (b) est un plan infini dont on filtre les angles d'incidence avant le capteur. Le disque et le plan sont définis comme des objets lambertiens. Dans l'expérience 2, l'illumination varie avec θ et une étape de normalisation est nécessaire (c).

Dans une seconde expérience, on place un plan infini, blanc et lambertien. Les rayons peuvent venir de toutes les directions sur le capteur, mais au lieu de fournir les réponses angulaires complètes, on fournit des réponses angulaires avec des zéros partout sauf à l'angle d'incidence que l'on veut tester. On simule alors une source collimatée étendue. En changeant à chaque fois l'angle d'incidence de test, on reconstruit la réponse angulaire pas à pas. L'avantage de cette méthode est qu'elle fournit une réponse angulaire directement, il n'y a pas besoin de transformation de coordonnées. Il faut néanmoins normaliser les données car le filtrage en angle fait varier l'illumination selon θ . Cela vient de l'interpolation linéaire des réponses angulaires car le maillage en angle a un pas $\Delta\theta$ de 5° . Si le maillage était beaucoup plus fin cet effet serait négligeable. Les rayons peuvent être lancés avec un angle θ compris dans $[\theta - \Delta\theta, \theta + \Delta\theta]$ et φ compris dans $[\varphi + \Delta\varphi, \varphi + \Delta\varphi]$. L'illumination augmente lorsque θ augmente et les sous-pixels reçoivent plus de lumière qu'attendu. Le problème est schématisé sur la Figure 3.5 c). Le terme de normalisation est donné par :

$$N = \underbrace{2\Delta\varphi}_{(a)} \times \underbrace{f_{\mu\text{Lens}}^2 [\tan^2(\theta + \Delta\theta) - \tan^2(\theta - \Delta\theta)]}_{(b)} \quad (3.15)$$

où (a) est la portion de surface délimitée par $\varphi \pm \Delta\varphi$ et (b) est la fraction de disque délimitée par $\theta \pm \Delta\theta$.

Pour les deux expériences il est nécessaires de tirer un grand nombre de rayons pour reconstruire les réponses angulaires. En effet, dans l'expérience 1 il faut réussir à toucher la cible qui est visible dans un cône de 0.3° environ (disque de 5mm à 1m); dans l'expérience 2 il faut réussir à passer le filtrage en angle. On lance 2^{20} rayons par sous-pixel soit 2^{22} rayons au total pour chaque point de mesure.

La Figure 3.6 montre la réponse angulaire du sous-pixel A pour les 2 expériences de validation. Ces résultats sont calculés pour les mêmes angles d'incidence et sont normalisés à 1. Les 2 expériences donnent des résultats très proches de la réponse angulaire des simulations. On constate des différences dues à l'interpolation pour ré-échantillonner les résultats sur la même grille en angle, et potentiellement dues au passage entre les coordonnées (x,y) à (φ, θ) pour l'expérience 1.

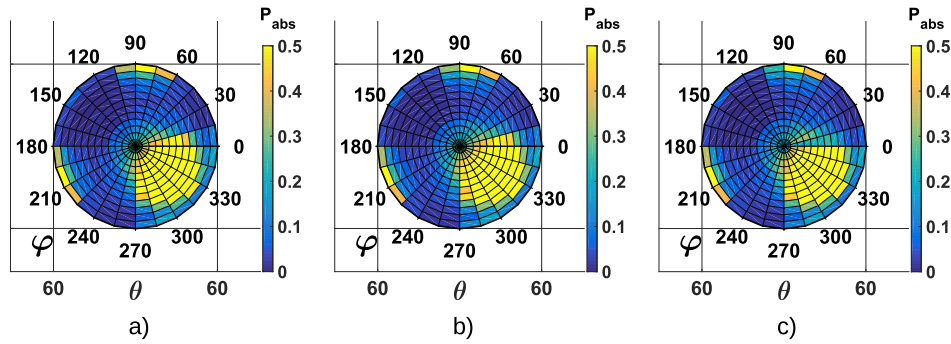


Figure 3.6. Comparaison des réponses angulaires : a) celles des simulations FDTD, b) Expérience 1, c) Expérience 2. **Note** : comme dans le chapitre précédent, il faut faire tourner le graphique d'un demi-pas (7.5°) dans le sens horaire à cause de la colorisation du maillage de Matlab.

3.4 Application à un capteur quad-pixel

3.4.1 Exemple d'images générées avec PBRT modifié

Dans cette partie, je présente des exemples d'images générées avec notre outil de rendu modifié. J'utilise un capteur de 5.95×5.95 mm avec des sous-pixels de $1.75 \mu\text{m}$ pour un total de 1700×1700 microlentilles. Les objectifs utilisés sont ceux décrits dans le brevet US 8,320,061 B2 (objectif grand angle) et US 9,316,810 B2 (télé-objectif avec un design de périscope), respectivement appelés *ML-1* et *ML-2* par simplicité. Ils sont représentés dans la Figure 3.7 et sont destinés à être des objectifs pour téléphones. Le CRA est corrigé sur le capteur pour ces 2 objectifs. On lance 128 rayons par sous-pixels pour obtenir une bonne qualité d'image. Dans cette partie, les images présentées ne sont pas dématricées : en effet PBRT fournit directement des images RGB. On utilise plusieurs mires différentes (ABCD, ou l'ISO-12233) en tant que textures planes pour tester le rendu (Figure 3.7c)). On utilise aussi des scènes 3D, principalement San Miguel pour avoir une scène avec différents objets à différentes profondeurs (<https://www.pbrt.org/scenes-v2.html>). La scène San Miguel est une modélisation créée par Guillermo M. Leal Llaguno d'un hotel à San Miguel de Allende, au Mexique. Elle est assez riche en éléments (piliers, arbres, tables, etc) et on peut y déplacer la caméra virtuelle pour générer plusieurs images prises d'un point de

vue différent. Sur la Figure 3.8 d) la table au premier plan se situe à environ 3 mètres, et l'arrière plan à une dizaine de mètres. Les exemples sont présentés dans la Figure 3.8

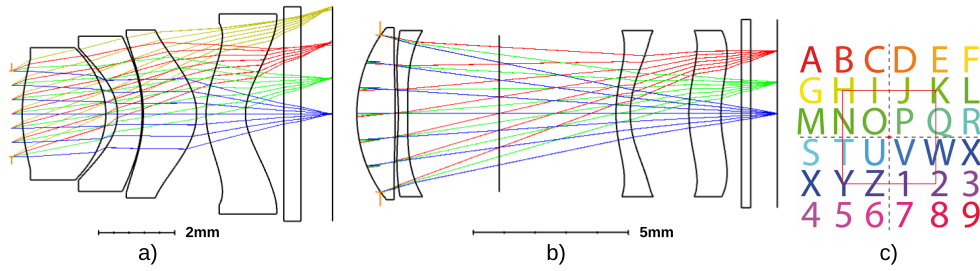


Figure 3.7. (a) ML-1, (b) ML-2, (c) ABCD texture.

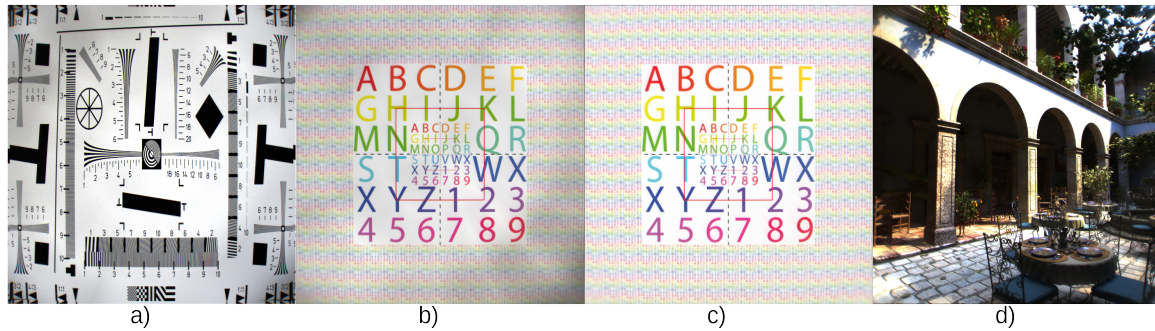


Figure 3.8. Exemples d'images de synthèse utilisant l'objectif ML-1. (a) et (b) montrent la SAI-A des 2 mires utilisées. (c) montre la SAI-A de la mire ABCD rendue avec les réponses angulaires pré-interpolées et corrigées par un "flat". (d) montre un point de vue de la scène San Miguel.

On constate des artefacts (des lignes lumineuses ou sombres) sur les images des SAIs. Elles sont dues à l'interpolation linéaire qui est utilisée pour calculer le poids du rayon à partir des réponses angulaires des simulations FDTD. Habituellement, pour un CRA constant (ou non corrigé) l'interpolation linéaire suffit car nos simulations électromagnétiques sont maillées relativement finement en angle. Cependant, le maillage en φ_{cra} et θ_{cra} est beaucoup plus large, ce qui entraîne ces artefacts lors de l'interpolation en 4D. Il est difficile de mailler φ_{cra} et θ_{cra} plus finement car chaque simulation FDTD est assez longue. Typiquement, nos simulations de quad-pixels $3.5\mu\text{m}$ prenaient entre 45 minutes et une heure selon la puissance de calcul disponible. Celles pour des quad-pixels de $0.8\mu\text{m}$ prenaient entre 10 et 20 minutes. De plus pour chaque point de mesure en CRA, il faut ensuite mailler en angle d'incidence φ et θ . Plusieurs solutions seraient envisageables pour répondre à ce problème sans forcément augmenter le nombre de simulations. On peut par exemple pré-interpoler les réponses angulaires en dehors de PBRT, et lui laisser faire une interpolation linéaire 4D lors du rendu pour éviter de trop dégrader les performances. On pourrait aussi tenter de trouver ou de définir une base de fonctions en 4 dimensions sur lesquelles faire une projection des réponses angulaires. On donnerait alors à PBRT les coefficients de cette décomposition. Enfin, on pourrait chercher un modèle analytique simplifié des réponses angulaires. J'ai choisi de pré-interpoler les réponses angulaires en utilisant la méthode "quadri-cubique". Cela permet de lisser légèrement les réponses, et l'interpolation linéaire de PBRT produit moins d'artefacts. J'utilise aussi une image "flat"

qui divise pixel à pixel chaque image générée. Cela permet de supprimer les variations de flux restantes. La différence est visible entre les Figure 3.8 b) et c).

3.4.2 Impact de la diffraction sur les SAIs

Dans cette partie, on utilise l'objectif *ML-1* avec une mise au point à 750 mm du capteur. La scène est la mire ABCD placée à 500 mm. Le défocus est volontaire car il faut montrer les décalages entre les SAIs, mais il est relativement réduit car on veut quand même pouvoir discerner les lettres. On utilise des microlentilles de 3.5 μm et de 1.6 μm en mode classique puis en mode diffractif. Seule la microlentille de 3.5 μm a fait l'objet d'une analyse complète en CRA. Les réponses angulaires de celles de 1.6 μm ont seulement été calculées en CRA constant et nul. On montre seulement quelques centaines de pixels au centre du capteur sur la Figure 3.9, là où le CRA est nul.

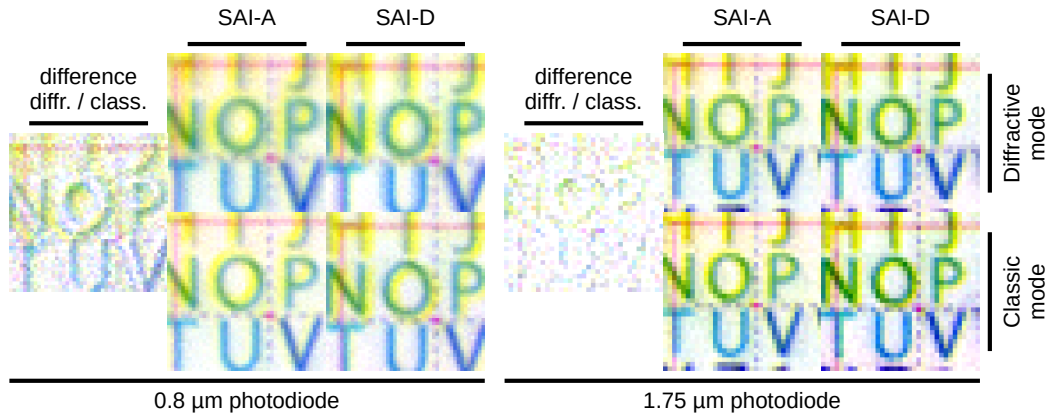


Figure 3.9. SAI A et D pour les sous-pixels de taille 0.8 μm et 1.75 μm , en mode classique et diffractif. La différence entre le mode classique et le mode diffractif est montrée sur la gauche à chaque fois.

Avec la microlentille de 1.6 μm les décalages sont encore visibles, ce qui prouve qu'un signal de disparités peut être extrait. Le crosstalk entre les SAIs reste très présent (visible sur la différence qui est assez marquée) et peut éventuellement mettre à mal les performances d'un autofocus par exemple. Au contraire, la microlentille de 3.5 μm est assez peu impactée par le crosstalk entre SAIs, la différence entre le mode classique et diffractif est très limitée.

3.5 Conclusion

Dans ce chapitre, je me suis appuyé sur les connaissances acquises lors des simulations d'un quad-pixel pour modéliser la formation des Images de Sous-Ouverture. Ce modèle lie d'une part l'optique utilisée pour produire l'image sur le capteur, et d'autre part la réponse angulaire des sous-pixels. Cette formulation pourra potentiellement servir à l'avenir si l'on veut optimiser conjointement l'optique principale et l'architecture des pixels. Avec la même notion de multiplication angulaire, j'ai modifié un outil de rendu qui permet maintenant de prendre en compte les effets de la diffraction dans les pixels. Cet outil est très important car il permet d'étudier les algorithmes de traitement d'images sans

forcément avoir un vrai capteur plénoptique quad-pixel (ni commercial, ni un prototype qui n'a été fabriqué que plus tard). Cet outil a permis de générer un dataset qui a servi de base à des entraînements de réseaux et neurones via des techniques d'apprentissage profond. C'est ce que j'aborde dans la partie suivante. Il permet aussi d'étudier l'influence de l'architecture d'un pixel plénoptique, typiquement la taille des sous-pixels, leur nombre ou les structures comme les [DTI](#). La visualisation peut se faire de manière plus directe, sur des images de mires ou de scènes 3D, au lieu de se faire sur des diagrammes de réponses angulaires. La méthode de rendu a été publiée dans notre article Optic Express de 2019[[16](#)] où l'on présentait au préalable les simulations FDTD du chapitre précédent.

Traitement de l'image

Le traitement d'image associé aux caméras plénoptiques et aux matrices de caméras est un domaine de recherche qui existe depuis déjà plusieurs années (Lytro, 2005 par exemple). Cependant l'apparition des capteurs plénoptiques de type 1 "minimalistes" et intégrés, le dual pixel en tête, est beaucoup plus récente (Canon 70D, 2013). Le traitement d'image associé est donc un sujet de recherche actuel et encore relativement peu abordé. On peut citer les développements du mode portrait (bokeh artificiel par segmentation du premier et de l'arrière plan, Figure 4.1) pour les smartphones par Google[27, 81, 88] à partir de 2018, l'extension de l'angle de vue par Vandame[78] en 2018 également, la suppression de réflexions[63] en 2020, ou l'extension de la profondeur de champ[1, 2] en 2020 et 2021 . Dans ce chapitre, je présente le travail réalisé sur les algorithmes de traitement d'images consacrés au quad-pixel.



Figure 4.1. *Isolation d'une personne et de l'arrière-plan en utilisant l'information de profondeur d'un dual-pixels d'un smartphone Google.*

En utilisant l'outil de rendu présenté dans le chapitre précédent, j'ai généré une base de données d'images que j'ai ensuite utilisée pour étudier différents algorithmes de traitement d'images. L'objectif est de concevoir une chaîne de traitement dédiée spécifiquement au quad-pixel, en utilisant à la fois des approches conventionnelles mais aussi basées sur des réseaux de neurones. Je présente dans un premier temps une méthode de correction des aberrations de l'objectif et je compare les performances selon la taille de la microlentille utilisée. Cela permet aussi de comprendre l'impact de la diffraction de la microlentille

sur les algorithmes de traitement d'image. Je présente ensuite plusieurs pistes d'utilisation de réseaux de neurones et de méthodes d'apprentissage profond pour le quad-pixel. La première concerne le dématricage couleur, la deuxième est une tentative de réduire le crosstalk entre les images de sous-ouverture (SAI), et la dernière est l'estimation passive de la profondeur. Ces expérimentations sont pour l'instant indépendantes mais peuvent potentiellement servir de briques de base pour un *pipeline* de traitement d'images plus global.

4.1 Base de données d'images

L'outil de rendu présenté au chapitre précédent a permis de générer plusieurs lots de 35 images de 3400 x 3400 pixels de côté, en utilisant un capteur quad-pixel de 3.5 μm et l'optique de smartphone décrite dans le brevet US 8,320,061 B2. Les microlentilles sont translatées pour corriger le CRA en suivant sa fonction CRA_{func} trouvée à partir du logiciel de conception Optics Studio (Zemax). Un lot comprend 23 images issues d'une scène 3D et 12 images placées comme texture plane. La mise au point de l'optique est faite à 1 mètre pour la majorité des scènes, cependant j'ai aussi fait quelques rendus avec une mise au point à 5 mètres sur la scène San Miguel. L'exemple présenté sur la Figure 4.2 c) est le seul cas particulier où la mise au point a été faite à 20 centimètres et 40 centimètres car la caméra est très proche de la table.

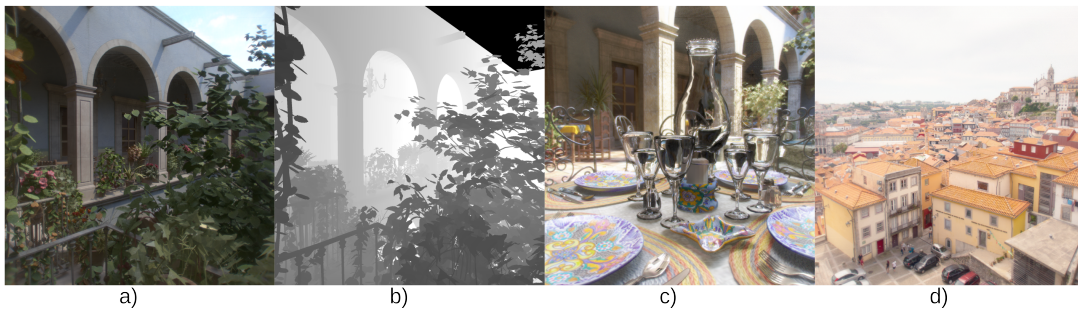


Figure 4.2. Exemples d'images de notre base de données. a) et c) sont issues de la scène San Miguel, b) est la carte de profondeur de a), et d) est une photo personnelle placée comme texture plane.

Les différents lots d'images correspondent à plusieurs paramètres différents. Le premier a été généré sans diffraction, c'est à dire en mode purement géométrique. Le second a été généré avec notre méthode de lancé de rayons modifiée et les réponses angulaires du quad-pixel de 3.5 μm . Dans ces deux cas, l'objectif est ouvert à F#2.2. Enfin, nous avons aussi généré les cartes de profondeur associées aux scènes 3D. Elles sont données directement en mètres et servent de vérité terrain pour l'entraînement des réseaux dédiés à l'estimation de profondeur. On peut les convertir en carte de disparités en utilisant l'Équation (1.5).

4.2 Rappels d'éléments en traitement d'images

Dans cette partie, je défini notations et je rappelle les notions que je vais utiliser au cours de ce chapitre pour les deux approches de traitements d'image. Les parties *block-matching*, *précision sous-pixellique* et *transformation locale* sont utiles pour comprendre la correction des aberrations. Les parties *convolutions discrètes* et *fonctions d'activation* expliquent les notions à la base des réseaux de neurones convolutifs. Enfin, la partie *fonctions de coût* est utile pour les deux approches.

4.2.1 Rappels sur l'algorithme de *block-matching*

L'algorithme de block-matching permet de mesurer un déplacement (dans notre cas, les disparités) entre deux images A et B, pixel par pixel. Le principe est le suivant : on extrait une sous-image P centrée autour d'un pixel p de l'image A. Cette sous-image est appelée *patch* en anglais, j'utiliserai ce terme par la suite. On déplace ce patch dans l'image B et pour chaque position, on calcule la valeur d'une fonction de coût qui estime la similitude entre P et le patch de l'image B à la même position. La différence entre les coordonnées du pixel p sur l'image A et les coordonnées où la fonction de coût est minimale (ou maximale) sur l'image B donne un vecteur de déplacement. On définit souvent une fenêtre de recherche W réduite et centrée sur les coordonnées du pixel p au sein de l'image B pour limiter les calculs et éviter de parcourir toute l'image B à chaque fois. Dans notre cas, il n'existe que des translations entre les SAIs. Ainsi, il n'est pas nécessaire d'ajouter des paramètres supplémentaires à la méthode ci-dessus, par exemple pour prendre en compte une éventuelle rotation ou un facteur de zoom.

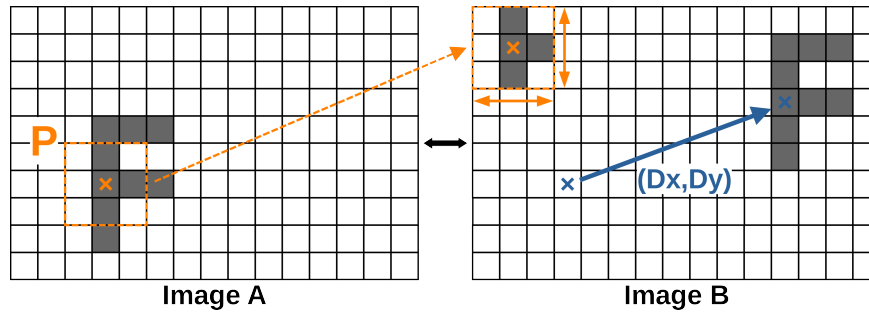


Figure 4.3. Illustration de l'algorithme de block-matching. Le patch P en orange est extrait de l'image A puis déplacé au sein de l'image B. Pour chaque position, la fonction de coût est calculée. La position minimisant ou maximisant la fonction choisie donne la valeur du déplacement (D_x, D_y) représenté en bleu sur l'image de droite.

4.2.2 Fonctions de coût

Différentes fonctions de coût existent et je présente les plus utilisées dans cette partie.

Une des plus utilisée est l'erreur moyenne quadratique (MSE) et sa version non moyennée (SSD) :

$$MSE = \frac{1}{N} \sum_{i=0}^N (A_i - B_i)^2 \quad (4.1)$$

$$SSD = \sum_{i=0}^N (A_i - B_i)^2 \quad (4.2)$$

Le **PSNR** est une mesure en dB communément utilisée pour comparer deux images, plus sa valeur est élevée plus les images sont similaires. Elle dépend de la **MSE** et de la valeur maximale d'un pixel (255 pour une image codée sur 8 bits, 1 pour une image normalisée) :

$$PSNR = 10 \log_{10} \left(\frac{\max^2}{MSE} \right) \quad (4.3)$$

On trouve aussi la somme des différences absolues (**SAD**) et sa version moyennée (**MAD**) :

$$SAD = \sum_{i=0}^N |A_i - B_i| \quad (4.4)$$

$$MAD = \frac{1}{N} \sum_{i=0}^N |A_i - B_i| \quad (4.5)$$

Enfin, la corrélation normalisée (**ZNCC**) est définie comme suit :

$$ZNCC = \frac{\sum_{i=0}^N (A_i - \mu_A) \times (B_i - \mu_B)}{\sigma_A \sigma_B} \quad (4.6)$$

avec μ_A et μ_B les moyennes des images A et B, et σ_A et σ_B leurs écarts types.

Les articles de revue de Brown[9] ou Huang[40] présentent des techniques de block matching plus avancées et d'autres fonctions de coût. Hirschmuller[38] compare plusieurs associations de block-matching/fonctions de coût. Il en ressort que la fonction de coût doit être adaptée selon la méthode utilisée. Cela est aussi valable dans le cas des réseaux de neurones et de l'apprentissage profond où l'on peut définir des fonctions de coût selon le problème à résoudre (par exemple la restauration d'images[89], ou l'estimation de profondeur monoculaire[15]...).

4.2.3 Précision sous-pixellique

La méthode de block matching définie précédemment évalue un déplacement au pixel près, mais il est possible de raffiner le déplacement. Au lieu de simplement garder la coordonnée où la fonction de coût est minimale, on peut sauvegarder ses valeurs pour les pixels voisins. On obtient alors des points avec trois coordonnées (D_x, D_y, F) . Par exemple, en faisant passer une parabole par ces points et en calculant analytiquement son minimum, on peut obtenir une précision sous-pixellique sur les déplacements.

4.2.4 Transformation locale des pixels

D'une manière très générale, il existe deux façons pour appliquer une transformation T (translation globale, rotation, fonctions plus complexes...) à une image. On peut calculer une projection de l'image de départ vers l'image d'arrivée : c'est le "mode" forward. Si la transformation est inversible, on peut partir de l'espace d'arrivée et lire les pixels de l'image de départ, c'est le "mode" backward.

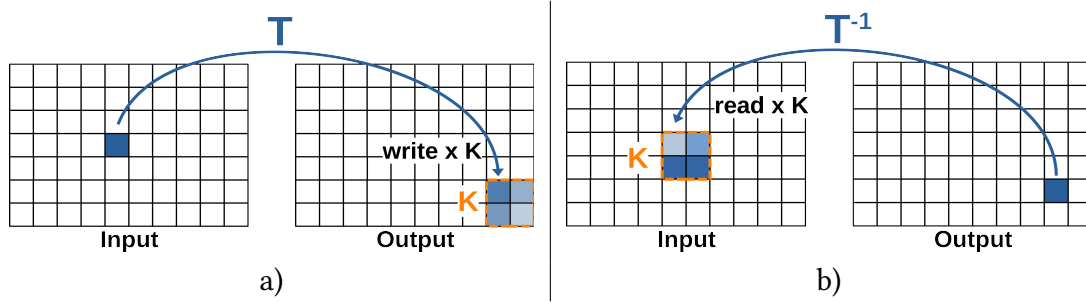


Figure 4.4. Interpolation effectuée en forward (a) et en backward (b).

4.2.4.1 Forward

En mode forward, on parcourt l'image d'entrée en calculant la transformation pour chaque coordonnée en pixels. On vient ensuite copier le pixel dans l'image d'arrivée depuis l'image d'entrée. On parle de splatting en anglais. Le splatting peut être réalisé au plus proche voisin (on copie juste le pixel à la bonne coordonnée arrondie) ou suivre une méthode d'interpolation (linéaire, cubique, ...) où l'on vient étaler la valeur du pixel d'entrée sur plusieurs pixels de l'image de sortie, avec un certain coefficient qui dépend de l'interpolation choisie et des parties décimales des coordonnées d'arrivée. Le problème de la méthode forward est qu'elle peut produire des pixels vides dans l'image d'arrivée, selon la transformation effectuée. De plus, le mode forward nécessite la gestion d'une carte de poids si l'on vient ajouter plusieurs valeurs à la même coordonnée de l'image d'arrivée. L'étape finale consiste à diviser l'image d'arrivée par la carte de poids pour assurer la bonne normalisation des valeurs.

4.2.4.2 Backward

En mode backward, il faut pouvoir inverser la transformation T . On parcourt l'image de sortie en calculant T^{-1} pour chaque coordonnée en pixels. On lit alors le pixel de l'image d'entrée aux coordonnées trouvées et on le place dans le pixel courant dans l'image d'arrivée. De cette façon, on empêche l'apparition de pixels vides car on parcourt toute l'image d'arrivée et la gestion d'une carte de poids est inutile. C'est donc cette méthode qui est choisie préférentiellement lorsque c'est réalisable. De la même manière que le forward, on peut travailler au plus proche voisin (on lit un seul pixel dans l'image d'entrée) ou on peut suivre une méthode d'interpolation (linéaire, cubique, Lanczos...) en lisant plusieurs pixels de l'image d'entrée et en en faisant la moyenne pondérée, où la pondération dépend de l'interpolation choisie.

4.2.4.3 Fonctions d'interpolation

La méthode d'interpolation dépend de la partie fractionnaire des coordonnées trouvées après le calcul de la transformation T ou T^{-1} . La valeur du pixel est multipliée par un noyau K dont la taille est variable selon la méthode considérée[34] (Figure 4.5). On peut prendre un exemple sur une seule dimension avec $X = E + F$, X la coordonnée après transformation, E sa partie entière et F sa partie décimale. Au plus proche voisin, on se contente d'utiliser E ou $E + 1$ selon l'arrondi. L'interpolation linéaire donne un noyau de

taille 2 défini par $K = [1 - F, F]$. Pour une interpolation de Lanczos, le noyau a quatre coefficients définis par $K_i = \text{sinc}(F)\text{sinc}(F/a_i)$ avec $a_{i=1,2,3,4} = \{-1, 0, 1, 2\}$. Pour passer en deux dimensions, on peut calculer ces noyaux pour les deux dimensions séparément puis faire leur produit tensoriel pour obtenir une matrice de coefficients (4 dans le cas linéaire, 16 avec Lanczos par exemple).

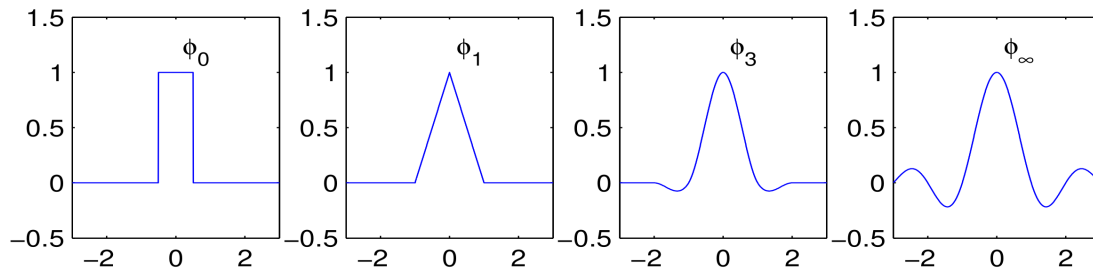


Figure 4.5. Exemple de kernels d'interpolation, dans l'ordre : plus proche voisin, linéaire, cubique et sinus cardinal. Illustration tirée de Boomgaard[77].

4.2.5 Limite du block matching dans le cas du quad-pixel

La variation de la PSF entre les images de sous-ouverture peut dégrader la précision de l'algorithme de block-matching. On peut comparer deux cas simples : un système stéréo composé de deux caméras identiques placées côte-à-côte, et un capteur plénoptique quad-pixel idéal. Si on considère une source ponctuelle légèrement en défocus, alors le système stéréo va produire deux images qui seront translatées entre elles (Figure 4.6 a). Le capteur quad-pixel va produire quatre images de sous-ouverture qui sont en fait des quarts de disque (Figure 4.6 b), on a représenté uniquement les deux SAIs supérieures). La ligne rouge représente le milieu de la scène : l'axe de la source est au milieu des deux caméras dans le cas du système stéréo, il est centré sur le capteur dans le cas du quad-pixel.

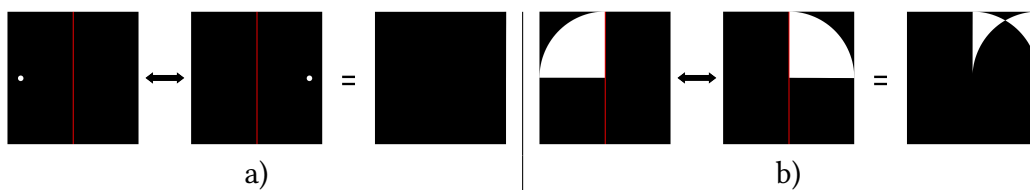


Figure 4.6. Comparaison de l'application de l'algorithme de block matching sur un système stéréo (a) et sur un quad-pixel (b). On a représenté uniquement les deux SAIs supérieures. Dans le premier cas il est possible de trouver une seule valeur de translation qui annule la fonction de coût. Les PSF en miroir du quad-pixel ont tendance à empêcher cette unicité car il reste toujours un résidu, quelle que soit la translation.

Lorsqu'on applique l'algorithme de block-matching au premier cas, on trouve une seule position qui minimise la fonction de coût car la PSF est la même pour les deux caméras. En revanche les PSF en miroir des SAIs peuvent empêcher l'unicité du minimum de la fonction de coût, d'où la dégradation de la précision. C'est d'autant plus vrai si on considère un quad-pixel réel, avec du crosstalk entre les SAIs (Figure 3.2), au lieu d'un

quad-pixel parfait. On a le choix de comparer deux SAIs entre elles, ou alors de comparer une SAI par rapport à la somme des quatre SAIs. Je n'ai pas comparé la précision des deux approches et dans la suite de ce chapitre, j'ai utilisé le block matching entre une SAI et leur somme.

4.2.6 Convolutions discrètes

Une convolution permet de filtrer une image par un noyau de convolution (kernel) et d'obtenir des fonctions variées, comme une réduction du bruit ou la détection de contours. Si l'on appelle I_e l'image en entrée, I_f l'image filtrée et K le filtre de taille s , les pixels de I_f sont calculés suivant :

$$I_f(i, j) = \sum_{k=0}^{s-1} \sum_{l=0}^{s-1} I_e \left(i - \left\lfloor \frac{s}{2} \right\rfloor + k, j - \left\lfloor \frac{s}{2} \right\rfloor + l \right) \times K(k, l) \quad (4.7)$$

où $[x]$ est la partie entière de x . Dans le cas d'un réseau de neurones convolutif, les poids de chaque convolution est un paramètre libre que le réseau met à jour à chaque itération. De plus, le pas de déplacement du noyau, que l'on appelle *stride* en anglais, peut être variable. L'équation ci-dessus correspond à un stride égal à 1, mais il est possible de déplacer le noyau avec un stride plus élevé. Cette technique est par exemple utilisée dans un réseau de neurones pour diminuer la résolution spatiale d'une image d'un facteur 2 dans les deux directions, en utilisant un kernel de taille 2 et un stride de 2. En effet quatre pixels de l'image d'entrée contribuent à un seul pixel de l'image filtrée. **Note :** il existe des méthodes non convolutives pour réduire les dimensions d'une image, par exemple le *max-pooling* ou l'*average-pooling* où l'on prend le maximum ou la moyenne d'un patch de pixels.

4.2.7 Fonctions d'activation

Les réseaux de neurones entièrement convolutifs appliquent des fonctions d'activation, après chaque convolution, à tous les pixels. Cela permet d'apporter des non-linéarités dans l'architecture du réseau, une convolution seule étant linéaire. Ces non-linéarités permettent au réseau de changer la représentation des données et au final de réaliser des fonctions complexes[75].

La plus connue, et celle que j'ai le plus utilisé, est la fonction ReLU (Rectified Linear Unit) définie par :

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (4.8)$$

Si l'on souhaite estimer des disparités, il faut que les valeurs des pixels puissent être positives ou négatives. On peut alors utiliser des fonctions comme la tangente hyperbolique ou sa version linéarisée "*hard tanh*" :

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.9)$$

$$\text{hardTanh}_{m,M}(x) = \begin{cases} M & \text{si } x > M \\ -m & \text{si } x < -m \\ x & \text{sinon} \end{cases} \quad (4.10)$$

4.3 Correction des aberrations

4.3.1 Principe

Dans cette section, je présente un algorithme qui vise à corriger les aberrations de l'objectif principal. Habituellement, si un objet est en dehors de la zone de netteté, son image n'a pas la même position sur chaque SAI. Ce décalage est la disparité ρ et les algorithmes plénoptiques sont basés sur cette disparité[55].

Cependant, même si un objet est dans la zone de netteté, des disparités peuvent apparaître à cause des aberrations de l'optique principale (Figure 4.7). On peut mesurer ces disparités puis les corriger, ce qui améliore la qualité des images. Les gains dépendent de la nature des aberrations, certaines se corrigent mieux que d'autres, et aussi du nombre de sous-pixels par microlentille. Plus ils sont nombreux, plus on peut corriger des aberrations complexes. La correction est utile (et possible) seulement si le spot diagram est plus grand qu'une microlentille.

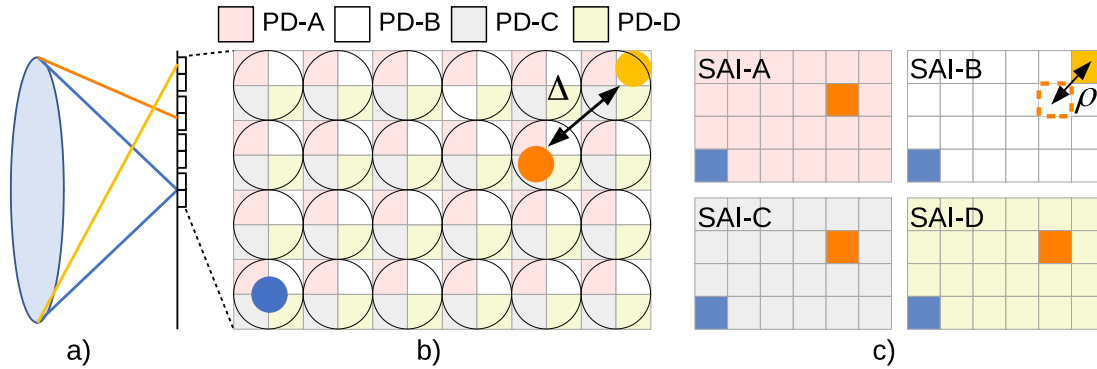


Figure 4.7. Aberrations de l'optique principale vues par un capteur quad-pixel. (a) montre une aberration sur un coin du capteur où les rayons orange et jaunes ne convergent pas au même point, contrairement aux rayons bleus au centre. Les figures b) et c) représentent les microlentilles et les SAIs du quart supérieur droit du capteur. b) montre l'impact des rayons sur les microlentilles. c) montre les quatre SAIs et l'impact des rayons sur les sous-pixels. L'écart de position Δ sur les microlentilles se traduit par des disparités ρ sur les SAIs.

La méthode de correction consiste à prendre une photo d'un objet fortement texturé (typiquement une mire de bruit, Figure 4.8) et parfaitement mise au point. On utilise ensuite la technique de *block-matching* pour calculer les disparités entre les SAIs et une référence. Cette référence peut être une unique SAI, ou bien leur somme. On obtient une carte de translation ΔX et ΔY par pixel, par SAI et par couleur. On transforme chacune des SAIs en translatant leurs pixels de la valeur opposée de la carte des translations. Dans notre cas, la translation est opérée par une interpolation de Lanczos en mode forward.

Cette correction permet d'améliorer la qualité des images[8], moyennant une acquisition d'une mire de bruit. Elle est également nécessaire lors de l'estimation de la profondeur.

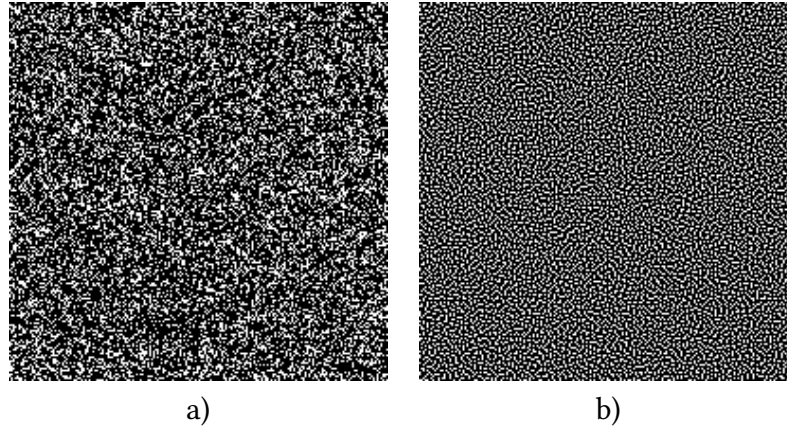


Figure 4.8. Mire de bruit blanc à gauche, bruit bleu à droite. La couleur du bruit fait référence à ses fréquences spatiales. Les fréquences spatiales d'un bruit blanc ont toutes la même énergie, tandis que pour un bruit bleu les hautes fréquences sont plus présentes que les basses fréquences. Illustrations tirées de <http://momentsingraphics.de/BlueNoise.html>.

En effet, il faut isoler les disparités dues à la profondeur de celles dues aux aberrations[81].

4.3.2 Comparaison des performances selon le niveau de diffraction

J'ai appliqué cet algorithme sur des images issues d'un quad-pixels de $1.6\ \mu\text{m}$, au centre et dans les coins. Les réponses angulaires du quad-pixel $1.6\ \mu\text{m}$ sont décrites dans le chapitre 3 (Section 2.1.4). Comme le CRA n'a pas été entièrement simulé pour ce pixel (contrairement au $3.5\ \mu\text{m}$), je simule seulement des portions de capteurs de 150×150 microlentilles en considérant que le CRA est constant. Le coin correspond à un CRA de $\theta_{cra} = 10^\circ / \varphi_{cra} = 45^\circ$. Trois cas sont testés : l'objectif 1 (US8320061 B2) sans diffraction dans les pixels, l'objectif 1 avec diffraction dans les pixels, et enfin l'objectif 2 (US9316810 B2) avec diffraction dans les pixels. L'objectif 2 a été, en plus, dégradé par rapport à la description du brevet. Cela a pour but de simuler des défauts de fabrication par exemple. Les défauts sont résumés dans le Tableau 4.1. J'utilise la ZNCC pour la fonction de coût et un fit parabolique pour obtenir des déplacements décimaux.

Élément	Erreurs
1	$\Delta z = 1.5\ \mu\text{m}$
2	$\Delta x = 6\ \mu\text{m}, \Delta y = 6\ \mu\text{m}$
miroir	tip = 0.625° , tilt = 0.25°
3	$\Delta x = 9\ \mu\text{m}, \Delta y = 5\ \mu\text{m}$, tip = 0.2° , tilt = 0.2°
4	$\Delta x = 4\ \mu\text{m}, \Delta y = 2.5\ \mu\text{m}$

Tableau 4.1. Simulation de défauts de fabrication pour l'objectif 2.

Les résultats sont regroupés dans la Figure 4.9. Les gains en netteté sont donnés dans le Tableau 4.2, en utilisant les mesures proposées par De[20] et par Wee [13], en plus de la moyenne des gradients dans les deux directions. Les 2 objectifs sont très bons au centre donc les gains sont quasiment invisibles quel que soit le cas simulé. On observe par contre des différences entre la simulation avec diffraction et celle sans diffraction pour l'objectif 1.

Dans le cas non diffractif, il apparaît que les gains sont meilleurs que dans le cas diffractif. Cela est confirmé par les mesures, mais aussi visuellement sur la figure (zones entourées en rouge par rapport aux zones entourées en bleu).

En mode géométrique, les réponses angulaires sont des quarts de disque bien définis et donc le crosstalk entre les sous-pixels n'est pas présent. Le calcul des disparités est facilité et l'on ramène des sous-images relativement nettes au même endroit ce qui donne une somme assez nette elle aussi. En mode diffractif, le crosstalk entre SAIs rend plus difficile l'estimation des disparités, et en plus chaque SAI comporte une certaine trace des autres SAIs. On ramène des sous-images relativement floues au même endroit, donc la somme est dégradée.

Cela dépend aussi de la complexité des aberrations. Par exemple, les défaut de fabrication simulés sur l'objectif 2 semblent mieux corrigés que les aberrations de l'objectif 1. Cela est visible sur la Figure 4.9 avec les lettres entourées en vert, ou via les mesures du Tableau 4.2. Ces corrections sont intéressantes pour deux aspects. Elles pourraient permettre de relâcher les contraintes de fabrications des objectifs (alignement entre les éléments) et donc réduire les coûts, surtout si le volume de production est très élevé (typiquement le marché des smartphones). Elles permettraient aussi une démarche de co-conception entre l'objectif et le traitement d'image : on optimiserait le design optique uniquement pour les aberrations non corrigées. Celles bien corrigées par le capteur seraient laissées de côté. Cela pourrait permettre de diminuer la complexité du design, ou diminuer le nombre de lentilles asphériques par exemple.

	mesure	Centre			$\theta_{cra} = 10^\circ / \varphi_{cra} = 45^\circ$		
		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 [20]	0.2378	0.2384	+0.25%	0.0703	0.0875	+25.5%
	Wee [13](E^{-12})	0.5700	0.5811	+1.96%	0.3415	0.4027	+17.9%
<i>ML-1</i> diffr.	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> diffr.	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%

Tableau 4.2. Gains en netteté selon plusieurs mesures.

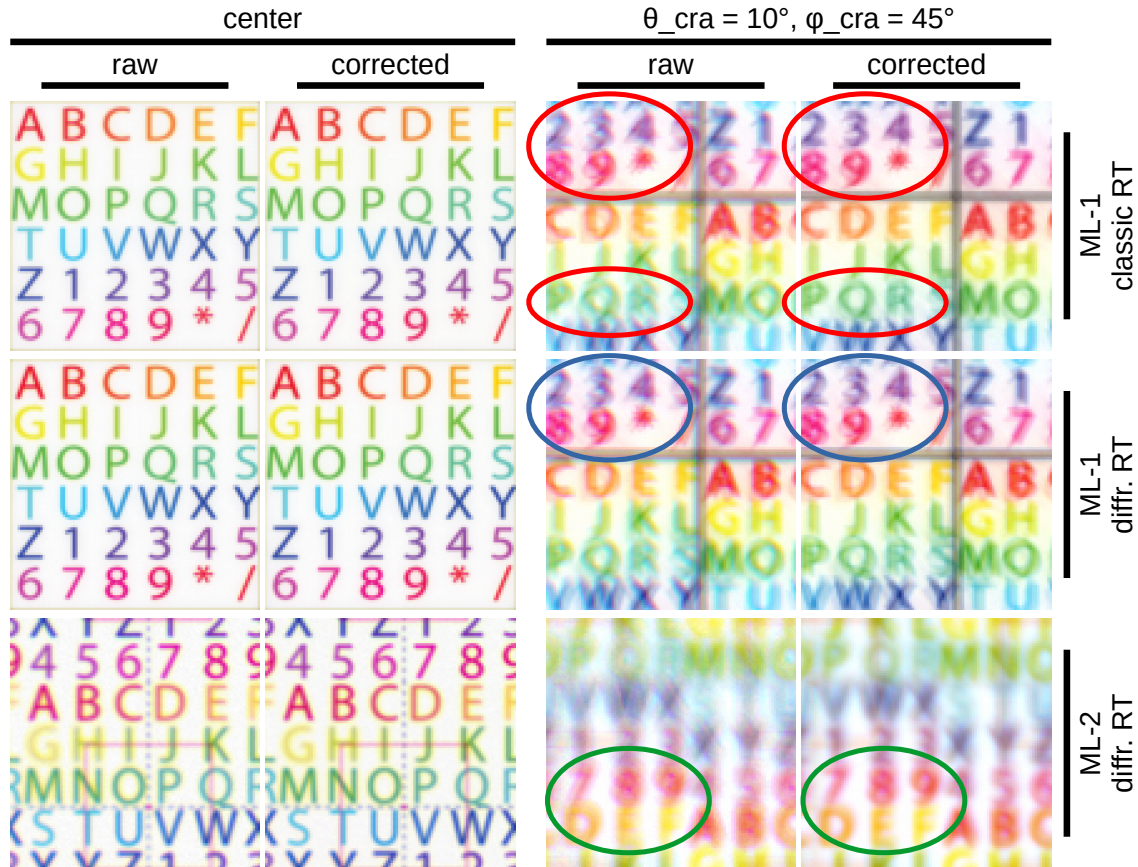


Figure 4.9. Résultats de la correction des aberrations pour une zone au centre et dans un coin ($\theta_{cra} = 10^\circ / \varphi_{cra} = 45^\circ$). Trois cas sont présentés : objectif 1 sans diffraction (ligne 1), objectif 1 avec diffraction (ligne 2) et objectif 2 avec diffraction (ligne 3).

4.4 Apprentissage profond

Dans les parties qui suivent, je présente mon travail sur des réseaux de neurones. J'aborde d'abord le dématricage couleur, puis la correction de la diffraction et enfin l'estimation passive de la profondeur. Comme on souhaite traiter des images en entrée et obtenir des images en sortie, seuls les réseaux de neurones entièrement convolutifs [50, 6] (FCN) sont adaptés. Ils permettent de traiter les images pixels à pixels, contrairement aux réseaux convolutifs "simples" développés par Le Cun [44], qui prennent une image en entrée et fournissent une probabilité en sortie. Le principe est le suivant : on présente en entrée du réseau une image que l'on veut traiter, ou dont on veut extraire de l'information. Un certain nombre de convolutions, qui ont chacune une taille de kernel et un nombre de canaux de sortie, sont calculées. Les images en sortie sont appelées inférences et on les compare avec une image dite "vérité terrain" (*ground truth* en anglais). La comparaison s'effectue via le calcul d'une fonction de coût. Chaque convolution possède des poids (ou paramètres libres) qui sont mis à jour en calculant la dérivée de la fonction de perte par rapport à tous les poids du réseau. Dans les cas simples, on utilise une descente de gradient à variables multiples. En répétant l'opération un certain nombre de fois sur plusieurs images différentes, le réseau peut arriver à faire converger ses poids de manière à ce qu'il réponde comme on le désire [65, 22, 41]. On peut s'en servir pour calculer des inférences

sur d'autres images, dont on ne connaît pas la vérité terrain.

J'ai utilisé le langage de programmation Python et la librairie Pytorch pour développer et entraîner les réseaux qui suivent. Les calculs ont été réalisés sur une carte vidéo *Nvidia 1080 Ti* dotée de 11Go de mémoire. Le traitement préalable des données a été fait avec les modules classiques de Python (Numpy, Scipy) et openEXR pour ouvrir les rendus de PBRT.

4.4.1 Dématriçage

4.4.1.1 Principe

Dans le cas d'un capteur plénoptique de type 1, les filtres de couleur sont placés par microlentille en suivant le motif de Bayer. Cela signifie que les SAIs présentent une mosaïque de couleur et doivent être dématriçées, c'est-à-dire que l'on doit trouver les 2 couleurs manquantes pour chaque pixel. Il existe déjà des réseaux de neurones pour le dématriçage des images RGB, par exemple le réseau présenté par Syu[74] dans sa version simple ou profonde (Figure 4.10 a) et b) ou le réseau présenté par Verma [80] (Figure 4.11). Le réseau de Syu dans sa version simple est composé de trois convolutions et son architecture ne présente pas de particularité. La version profonde est un réseau résiduel global : l'image est avant tout dématriçée de manière bilinéaire, puis le réseau corrige les erreurs résiduelles. Le réseau de Verma est basé sur des blocs résiduels en goulot d'étranglement[36] (residual bottleneck en anglais).

Dans notre cas, ces réseaux doivent être appliqués à chaque SAI indépendamment et les 4 SAI dématriçées doivent être additionnées pour produire l'image finale. Cependant, les SAIs peuvent être globalement décalées entre elles pour refocaliser l'image sur un autre plan de netteté, ou localement décalées pour corriger les aberrations de l'objectif comme on l'a vu dans la partie précédente. En fonction du décalage appliqué, il est possible d'obtenir naturellement une image finale partiellement dématriçée. Par exemple, si le paramètre de refocalisation ρ est 0, on doit dématriçer entièrement chaque SAI, mais si $\rho = 1$ alors l'image finale est complètement dématriçée automatiquement. En effet dans ce dernier cas, les 4 pixels des 4 couleurs différentes (en comptant les deux verts séparément) se retrouvent superposés au même endroit.

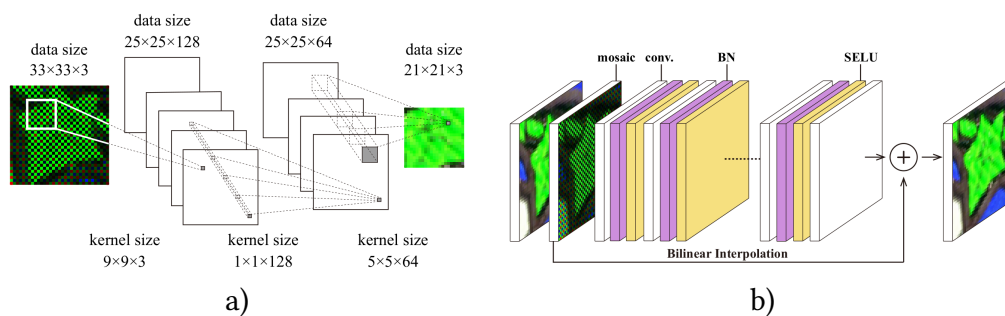


Figure 4.10. Les réseaux proposés par Syu et. al, en version simple à gauche et profonde à droite. Illustration tirée de Syu[74].

Dans cette partie, je compare les réseaux présentés ci-dessus utilisés 4 fois indépendamment sur les SAIs, à notre réseau qui prend douze canaux en entrée (les quatre SAIs

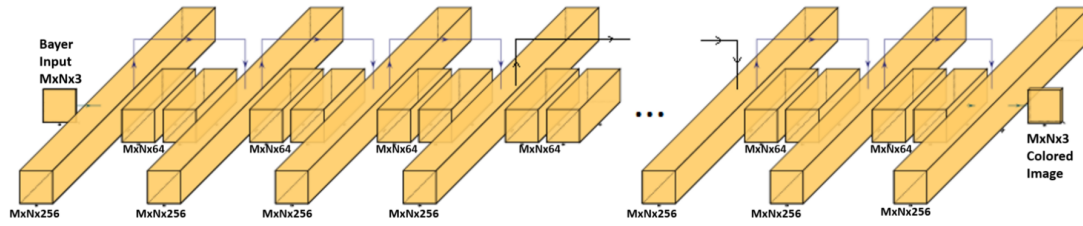


Figure 4.11. Le réseau proposé par Verma et. al, formé par 20 blocs résiduels. Illustration tirée de Verma[80].

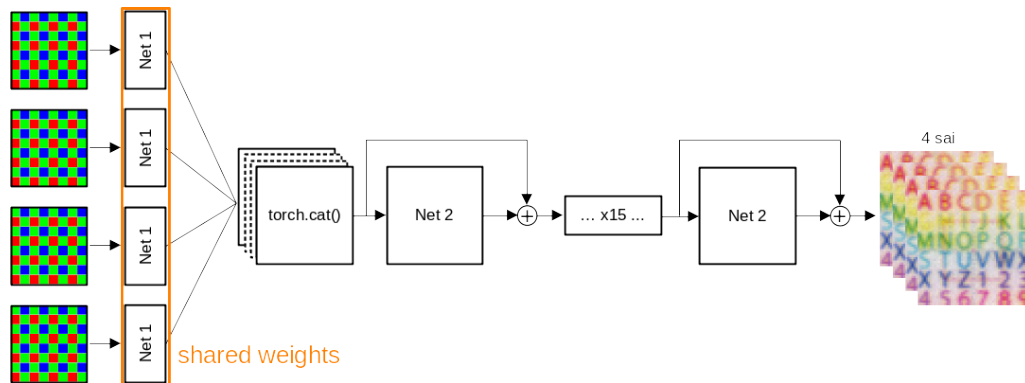


Figure 4.12. Architecture de notre réseau de dématricage.

matricées et concaténées) et fournit douze canaux en sortie (4 SAIs dématricées). Il est également basé sur des blocs résiduels (Figure 4.12), car cette architecture semblait meilleure que les réseaux plus classiques, et dispose de 485k paramètres (Syu VD : 670k, Verma : 714k). Chaque SAI passe par un premier réseau (Net 1) composé d'une première convolution avec un kernel de taille 7 et 48 canaux, puis deux autres convolutions avec un kernel de taille 5 et 64 canaux, chacune suivie d'une fonction d'activation ReLu. Ce premier réseau est partagé entre les SAIs et leurs résultats sont concaténés. Les 15 blocs résiduels (Net 2) sont similaires à ceux de Verma : une convolution avec un kernel de taille 1 et 128 canaux, une convolution avec un kernel de taille 3 et 32 canaux et une convolution avec un kernel de taille 1 et 128 canaux. Là encore, elles sont suivies d'une fonction ReLU. Une dernière convolution est nécessaire pour passer de 128 canaux en sortie du dernier bloc résiduel aux 12 canaux souhaités et on choisit un kernel de taille 1.

4.4.1.2 Entraînement

Pour entraîner le réseau, j'ai défini une fonction de coût modifiée. Je calcule la MSE de chaque SAI par rapport à sa vérité terrain, mais aussi la MSE de l'image finale par rapport à sa vérité terrain. L'objectif est de faire en sorte que le réseau apprenne l'image finale, mais aussi d'avoir des SAIs de bonne qualité si nécessaire. Si l'on supprime la somme des MSE des SAI, certaines expériences ont montré une image finale très bonne mais des SAIs présentant des artefacts, ce qui les empêche d'être utilisées ultérieurement. Ces deux facteurs peuvent être pondérés différemment avec $\beta = 1 - \alpha$. J'ai utilisé $\alpha = 0.5$ et $\beta = 0.5$.

$$L = \frac{\alpha}{4} \sum_{s=SAI} \text{MSE}(s, \hat{s}) + \beta \text{MSE}(\text{FI}, \widehat{\text{FI}}) \quad (4.11)$$

Les images sont transformées en sous-images (patches) qui sont mosaïquées à l'aide d'un motif de couleur choisi au hasard par microlentille (RGGB, BGGR, GRBG, GBRG). On vient simplement placer des zéros aux endroits où la couleur est censée être inconnue, cela correspond physiquement aux filtres colorés placés sur le capteur. Le fait de choisir aléatoirement parmi plusieurs motifs permet d'étendre virtuellement la base de données : un triplet (R,V,B) issu du rendu sert plusieurs fois (une chance sur trois d'utiliser le rouge, le vert et le bleu). Enfin, les patches sont transformés en SAI et concaténés, formant 12 canaux de données (4 SAI de 3 canaux). On a également implémenté des transformations aléatoires comme des symétries ou des rotations, là encore pour étendre virtuellement la base de donnée. Une transformation importante pour ce problème est le décalage aléatoire entre les SAIs. J'ai choisi d'utiliser des valeurs entières comprises entre -2 et +2 pour les axes horizontaux et verticaux. Les décalages ne suivent pas forcément l'équation de refocalisation car on veut aussi prendre en compte les décalages locaux dus à la correction d'aberration par exemple. J'ai utilisé des valeurs entières car l'interpolation nécessaire à la translation en valeur décimale dégradait trop la vérité terrain.

Les réseaux Syu et Verma sont entraînés à l'aide de la base de données Flickr500[74] (500 images téléchargées depuis le site Flickr, ce dataset a été construit par Syu), tandis que notre réseau est entraîné sur la base de données personnalisée présentée en début de chapitre. On a sélectionné 5 images comme images de validation (2 scènes 2D, 3 scènes 3D) et 5 autres images comme images de test (2 scènes 2D, 3 scènes 3D). Ces 10 images ne sont pas vues par le réseau lors de l'entraînement. La taille des sous-images étant de 48x48 pixels, on en forme 21025 pour l'entraînement et 4205 pour la validation. Les images dédiées au test ne sont pas découpées et sont passées en entier au réseau.

Dans les résultats présentés dans la section suivante, je donne la moyenne du PSNR des 5 images de test uniquement. Cela devrait limiter l'impact du biais dû à l'utilisation de base de données différentes. Le jeu de données Flickr500 est composé d'images très nettes, contrairement à notre base de données. Nous sommes principalement limités par les modèles 3D et les textures utilisées dans les scènes proposées de PBRT v2. D'une manière générale, les réseaux de Syu et Verma ont appris des cas plus difficiles (images très nettes et colorées) et cela devrait compenser le fait que notre réseau soit entraîné directement avec la base de données PBRT.

4.4.1.3 Résultats

Les inférences sont faites pour 2 cas : paramètre de refocalisation $\rho = 0$ et $\rho = 1$, et l'on calcule le PSNR sur l'image finale pour chaque cas et réseau (Tableau 4.3). Les réseaux pour les images simples sont utilisés indépendamment sur chaque SAI. La Figure 4.13 montre un exemple d'inférence de notre réseau.

Comme prévu, le cas où $\rho = 1$ est bien meilleur car chaque pixel reçoit la contribution des quatre couleurs différentes, et le gain est presque de +3dB pour chaque réseau. Notre réseau est légèrement meilleur, surtout lorsque le $\rho = 1$. Cela signifie que le réseau peut utiliser les décalages entre les SAIs pour améliorer la qualité du dématricage. Il est également important de noter que notre réseau est environ 25% plus léger que le réseau de Verma, ce qui pourrait être utile pour des utilisations à faible consommation d'énergie

	Bilineaire	Syu VD	Verma	Nous
$\rho = 0$	35.00964	36.0144	37.651	37.775
$\rho = 1$	37.743	38.878	40.753	40.865

Tableau 4.3. PSNR des inférences pour un paramètre de refocalisation $\rho = 0$ et $\rho = 1$. L'interpolation bilinéaire est donnée à titre de comparaison.

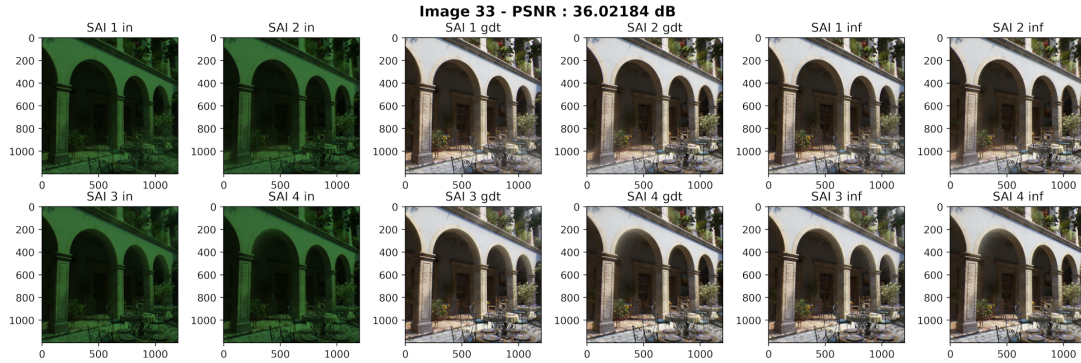


Figure 4.13. Exemple d'inférences en dématricage couleur (seules les SAIs sont représentées). À gauche : les SAIs mosaïquées, au centre : la vérité terrain, à droite : les inférences.

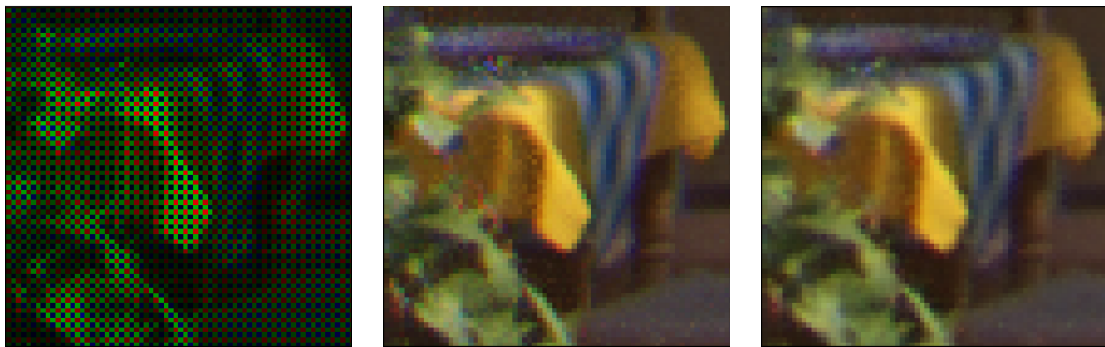


Figure 4.14. Inférence de la Figure 4.13 zoomée sur une zone au centre de l'image de sous-ouverture A. À gauche : l'image d'entrée mosaïquée, au centre : la vérité terrain, à droite : l'inférence du réseau.

comme les smartphones. Ces résultats doivent être confirmés car l'utilisation de 2 jeux de données différents pourrait avoir introduit un biais. Par exemple, on peut entraîner les réseaux Syu ou Verma sur le jeu de données PBRT au lieu du Flickr500. Je n'ai pas approfondi ce point, l'objectif étant plutôt d'évaluer les capacités des capteurs quad-pixel à fournir de l'information et à trouver des manières innovantes de la traiter.

4.4.2 Correction de la diffraction

4.4.2.1 Principe

Dans cette section, je présente une tentative pour corriger la diffraction causée par la microlentille, c'est-à-dire pour supprimer le crosstalk entre les SAI. J'ai utilisé deux archi-

tectures, un réseau résiduel similaire à celui du dématricage et un U-net (Ronneberger[68]) simplifié à 4 niveaux. Les images diffractives sont les images d'entrées, et les images sans diffraction représentent la vérité terrain. La diffraction apparaissant au niveau du pixel, j'ai décidé de travailler avec l'équivalent des images "brutes" de capteur : on applique le motif de la mosaïque couleur sur les images à l'aide d'un motif Bayer par microlentille, puis la dimension couleur est aplatie pour former un seul canal. Ces images sont découpées en sous-images et passées aux réseaux.

Le U-net suit l'architecture décrite par Ronneberger, mais nous n'utilisons que 4 niveaux (au lieu de 5) et 8 canaux pour le premier niveau (au lieu de 64). Sous cette forme allégée, le réseau a 120k paramètres et il est illustré par la Figure 4.16. Le réseau résiduel est très similaire à celui du dématricage mais moins profond, il est illustré par la Figure 4.15. Les fonctions ReLU suivent chaque convolution. Il a 382k paramètres :

- Une convolution **C1** de 256 canaux avec un kernel de taille 2 et un *stride* de 2. Elle a pour but de passer de l'image brute à une représentation en SAIs, mais une représentation apprise par le réseau, c'est-à-dire potentiellement avec des mélanges entre les SAIs.
- Trois blocs résiduels **resblock**. Chaque bloc est formé par 3 convolutions de 64, 64, et 256 canaux. Les kernel ont respectivement une taille de 1, 3, et 1.
- Une convolution transposée **CT** de 64 canaux, d'un kernel de taille 2 et un *stride* de 2. Elle est faite pour repasser de l'espace SAIs à l'espace "brut".
- Deux convolutions **C2** de 32 canaux et avec un kernel de taille 5.
- Une dernière convolution **C3** d'un seul canal.

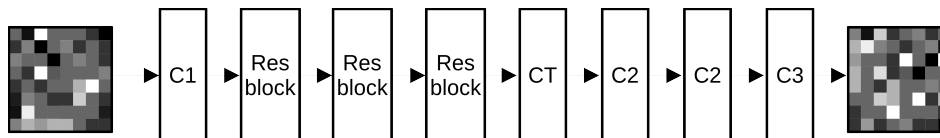


Figure 4.15. Réseau résiduel utilisé pour la correction de la diffraction.

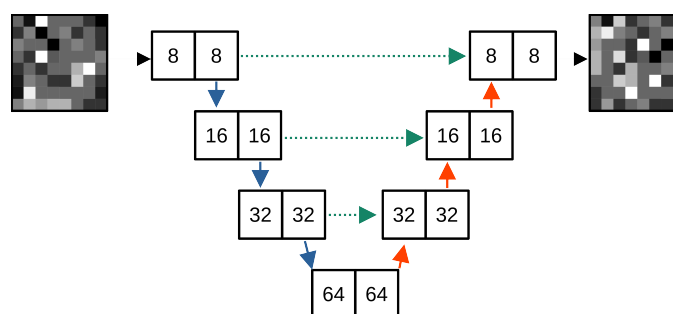


Figure 4.16. Réseau en "U" pour la correction de la diffraction. Les nombres de canaux sont indiqués dans chaque bloc. Les flèches en bleu sont des convolutions qui divisent la résolution spatiale par deux (noyau de taille 2, stride de 2). Les flèches en orange sont des convolutions transposées qui augmentent la résolution spatiale par deux. Les flèches en vert sont des connexions directes + concaténations.

La MSE est utilisée sur les images brutes comme fonction de perte. Un exemple d'in-

férence est présenté sur la Figure 4.17 et une version dématricée bilinéairement, pour une meilleure visualisation, dans la Figure 4.18. Sur les cinq images de test on mesure un PSNR moyen de 21dB pour le réseau résiduel et de 19dB pour le U-net. Cela montre qu'il est probablement possible de supprimer l'effet de diffraction par les microlentilles et de générer des SAI plus propres, sans crosstalk.

Une étude plus approfondie doit être effectuée, par exemple avec une meilleure architecture de réseau, une fonction de coût différente ou un meilleur jeu de données. Cette expérience était relativement facile à mener car j'ai utilisé des images de synthèse et un outil de rendu, décrit au chapitre précédent, qui permet d'ajouter ou de couper facilement les effets de la diffraction. Il semble plus difficile (voire impossible) de créer des images sans diffraction avec un vrai capteur. Ce point est discuté dans la section suivante.

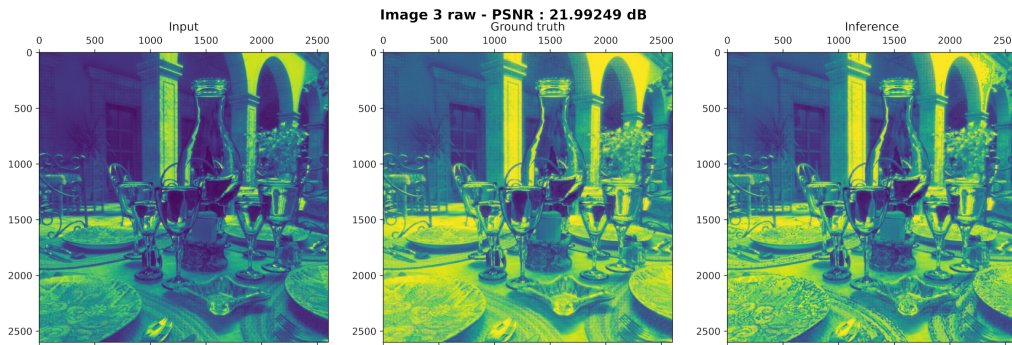


Figure 4.17. Exemple d'inférence sur une image "brute". L'image rendue avec diffraction est à gauche, c'est celle qui est passée au réseau. L'image rendue sans diffraction est au centre, c'est la vérité terrain. L'inférence du réseau est à droite. Le PSNR entre la vérité terrain et l'inférence est de 22dB.

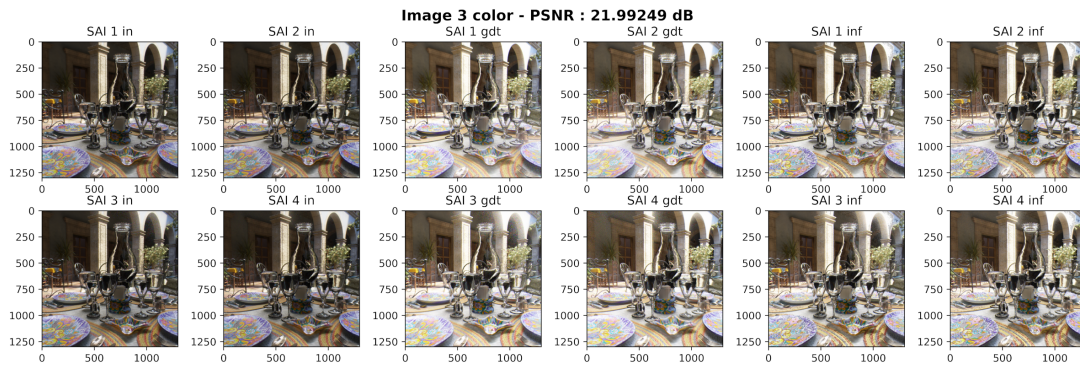


Figure 4.18. L'image brute de la figure Figure 4.17 a été dématricée avec une interpolation bilinéaire et présentée sous forme de SAIs. Elles sont placées dans le même ordre : image avec diffraction à gauche, vérité terrain au centre et inférence à droite.

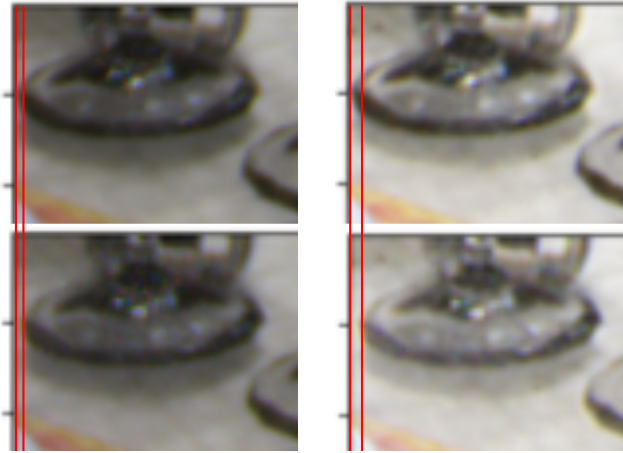


Figure 4.19. Zoom sur la figure précédente. La colonne de gauche représente les deux SAIs gauche et droite pour l'image avec diffraction. La colonne de droite représente les deux SAIs issues de l'inférence. Les lignes rouges sont tracées en prenant la partie gauche du pied du verre comme référence. L'écart entre ces lignes représente la disparité horizontale. On remarque qu'elle est plus marquée avec la diffraction corrigée.

4.4.2.2 Proposition de manipulation pour générer des images sans diffraction

Pour répondre au besoin d'un dataset sans diffraction dans la microlentille, on peut imaginer la procédure suivante (non réalisée expérimentalement) : on pourrait démonter un objectif et placer trois quarts de disque opaque au niveau de sa pupille de sortie. On s'attend à ce qu'un seul sous-pixel réponde, or la diffraction va provoquer du crosstalk dans les sous-pixels voisins (sous une même microlentille). Sachant quel quart de pupille est passant, on peut annuler les trois autres sous-pixels lors du traitement. En répétant l'opération quatre fois pour les quatre quarts de pupille, on pourrait virtuellement reconstruire une image sans diffraction en fusionnant les quatre acquisitions. La méthode est illustrée par la Figure 4.20. Il faudrait que l'objectif utilisé soit quasiment idéal et présente le moins d'aberrations géométriques possible. On pourrait éventuellement simplifier l'expérience en plaçant le masque devant l'optique et pas au niveau de sa pupille de sortie pour ne pas avoir à démonter l'objectif. Il faudrait quand même s'assurer que le cône de lumière est bien séparé en quarts avec un logiciel de conception optique. On peut imaginer compléter la mesure en plaçant des filtres colorés devant le capteur et ainsi produire des images en couleur à l'aide de 12 acquisitions. Avec trois acquisitions supplémentaires pour former une image avec diffraction, soit quinze au total, on pourrait former deux versions d'une même scène : avec et sans crosstalk entre les SAIs.

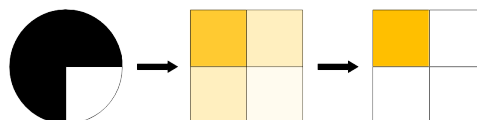


Figure 4.20. Illustration de la procédure pour générer des images sans diffraction.

4.4.3 Estimation passive de profondeur

Afin d'essayer d'estimer la profondeur de manière passive, j'ai ré-implémenté à la fois le DU2Net de Zhang[88] et le réseau de Punnappurath[62] (il s'agit du même article qui proposait une modélisation de la fonction de flou par SAI au chapitre 3), et je les ai adaptés au quad-pixel. Le Du2Net de Google est en fait un réseau prévu pour fusionner les données du dual-pixels et des doubles caméras de leurs nouveaux smartphones afin de produire une carte de profondeur. J'ai donc extrait uniquement la partie dual pixels du réseau. J'ai aussi tenté de ré-utiliser le Unet défini pour la correction de la diffraction, en passant d'une fonction d'activation ReLU à une tangente hyperbolique.

Lors de l'entraînement, les quatre SAIs en couleur sont placées en entrée du réseau et les cartes de disparités en tant que vérité terrain. Les cartes de disparités sont issues de la conversion des cartes de profondeur en utilisant l'Équation (1.5). C'est une étape nécessaire car certaines scènes ont été prises avec deux distances de mise au point différentes. Cela cause une indétermination car pour une même vérité terrain de profondeur (en mètres) il existe deux images possibles (mise au point à 100cm et 500cm). L'estimation relative des disparités en pixels autour du plan de netteté, au lieu d'une estimation absolue de la profondeur, lève cette indétermination. Dans un cas réel, pour repasser des disparités en pixels à une profondeur en mètre, cela nécessite de connaître la focale de l'objectif utilisé ainsi que sa distance de mise au point. C'est possible pour un appareil photo à objectif interchangeable mais plus délicat pour un smartphone (en général ces données ne sont pas enregistrées).

Malheureusement je ne suis pas parvenu à produire de bonnes cartes de profondeur, les différents réseaux n'ont pas convergé suffisamment, en particulier le réseau de Google et le Unet modifié.

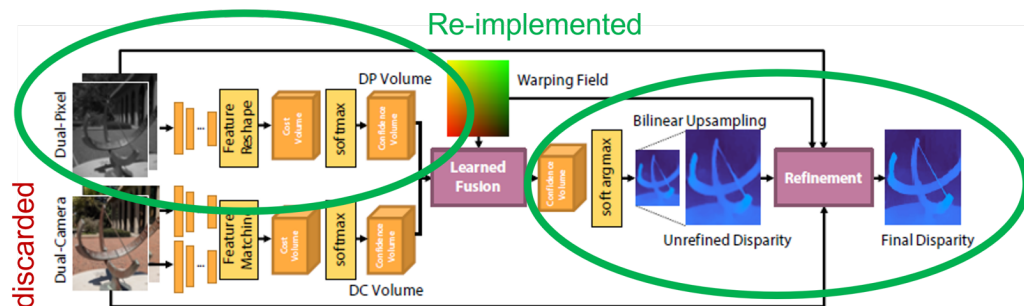


Figure 4.21. Réseau Du2Net. La partie dual-pixel a été modifiée pour prendre quatre SAIs au lieu de deux, la partie concernant la double caméra a été laissée de côté. Illustration tirée de Zhang[88].

Concernant le réseau de Punnappurath, la base de données que j'ai générée avec PBRT n'était pas appropriée car elle est basée sur des scènes 3D, alors que le réseau de Punnappurath est censé recevoir des patches avec une profondeur constante. Les seules images à profondeur constante disponibles sont les photographies utilisées comme texture planes, mais elles ont été générées à la même profondeur à chaque fois. Concernant le Du2Net, je pense que cela est également dû à notre base de données qui n'est pas assez grande pour entraîner correctement le réseau. De plus, les textures et les modèles 3D utilisés dans PBRT v2 ne sont peut être pas assez détaillés pour que le réseau «attrape» les disparités.

Cependant, l'estimation de profondeur passive est possible (Google y parvient) et elle est susceptible d'être meilleure que le dual-pixel car le quad-pixel apporte un axe supplémentaire pour calculer les disparités.

J'ai tout de même essayé d'appliquer la technique de block-matching sur les scènes San Miguel, en particulier la table (Figure 4.2c) et il semble qu'il existe des disparités (Figure 4.22) mesurables.

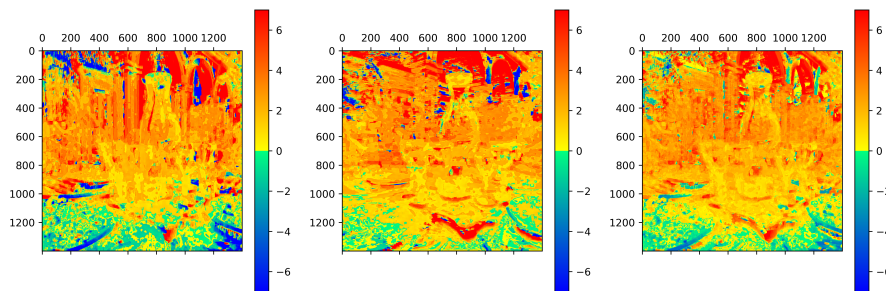


Figure 4.22. Exemple de carte de disparités calculées avec un algorithme de block matching. A gauche : la moyenne des disparités gauche et droite. Au centre : les disparités horizontales. A droite : les disparités verticales. L'unité de mesure est en pixels.

En particulier, on observe presque correctement l'avant-plan avec des disparités négatives (vert et bleu) et l'arrière-plan avec des disparités positives (orange et rouge). La scène de la table est particulièrement probante car les éléments sont entre 0 et 1 mètre, et le focus a été fait à 40 centimètres. Les résultats sont moins bons avec les scènes dont la mise au point est à un mètre. Une explication vient sans doute de l'objectif utilisé et de sa distance hyperfocale. C'est la distance de mise au point à laquelle une scène apparaît complètement nette. Or, contrairement à un système de caméras multiples, un capteur plénoptique de type 1 extrait les disparités du flou de défocus de l'objectif principal. Si ce dernier produit une image quasiment nette de toute la scène, il n'est plus possible de calculer les disparités car elles n'existent pas.

4.5 Conclusion

Dans ce chapitre j'ai utilisé l'outil de rendu présenté dans le chapitre précédent pour générer une base de données d'images qui a servi à étudier plusieurs algorithmes de traitement d'images. J'ai utilisé à la fois les approches classiques et à base de réseaux de neurones. L'objectif principal est de développer et d'évaluer de nouvelles possibilités d'utilisation d'un capteur quad-pixel, sans forcément chercher à optimiser finement chaque algorithme ni à dépasser l'état de l'art à chaque fois. En prenant l'exemple de la correction des aberrations, j'ai montré que la diffraction a un impact réel sur les algorithmes de traitement d'images. Pour une même taille de pixel, la correction est bonne pour des images sans diffraction mais dégradée pour les images avec diffraction. Concernant la correction elle-même, rendue possible par l'utilisation d'un capteur plénoptique intégré, elle peut servir de base à une conception conjointe avec le design optique de l'objectif. En

particulier, elle peut servir à relâcher des contraintes de fabrication ou même simplifier le design.

Les images générées ont aussi servi pour l'entraînement de réseaux de neurones convolutifs pour trois applications : le dématricage couleur, la correction de la diffraction et l'estimation de profondeur. Si le quad-pixel n'améliore pas énormément la qualité du dématricage, il permet de réduire la taille des réseaux. Cela peut être intéressant pour des applications où l'on veut économiser de l'énergie, par exemple pour les smartphones. L'apprentissage profond appliqué à la suppression de la diffraction donne des résultats intéressants et doit être poussé plus loin. Un réseau assez entraîné pourrait devenir la première étape du traitement en supprimant le crosstalk entre les images de sous-ouverture et ainsi aider les autres algorithmes utilisés par la suite. Concernant l'estimation de la profondeur, je n'ai pas pu reproduire les résultats des articles précédents sur le dual-pixel. Je pense que cela est principalement dû à la base de données qui est trop petite pour alimenter correctement des réseaux complexes. De plus, les scènes disponibles pour PBRT v2 peuvent ne pas être suffisamment détaillées et variées. Les modèles 3D sont peut-être pas assez précis, ou les textures pas assez détaillées. L'estimation des disparités à la main par block matching donne des résultats assez peu précis à cause du problème de PSF. Pour répondre à ce problème, nous avons réalisé un prototype de quad-pixel dans le but de constituer un jeu de vraies images.

La partie concernant la correction des aberrations a été publiée en partie dans l'article Optic Express[16], et celle concernant le modèle de la PSF puis les réseaux de neurones a été publiée dans un article de conférence SPIE[17].

Expériences

J'ai travaillé jusqu'à présent exclusivement à l'aide de simulations, que ce soit pour évaluer les performances à l'échelle du pixel ou générer des images de synthèse qui ont permis de développer des algorithmes de traitement d'images. Pour s'assurer de leur justesse et donc de leur pertinence, il faut les comparer au monde réel. Dans ce chapitre, je présente deux expériences qui ont pour but de comparer les simulations aux mesures réalisées sur un vrai capteur. La première, réalisée à Interdigital, concerne le dual-pixel : on mesure les réponses angulaires d'un Canon 5D mk IV que l'on peut comparer aux simulations faites par Kobayashi[42]. La seconde est la plus importante car elle se base sur un prototype de quad-pixel fabriqué par STmicroelectronics et par le CEA-Leti. Le prototype a été conçu spécialement pour être comparé à nos simulations. En effet, la taille des sous-pixels utilisés est de $1.75\text{ }\mu\text{m}$ et la puce a été divisée en plusieurs zones de test qui correspondent aux simulations effectuées. Je détaille le design du prototype dans la suite du chapitre puis j'explique sa mise en oeuvre et enfin je présente les résultats. Ces derniers comprennent à la fois les réponses angulaires des différentes zones, mais aussi des images prises avec le prototype.

5.1 Dual pixel Canon

Le Canon 5D mk IV possède un capteur dual-pixels (Chapitre 1) de $24 \times 36\text{ mm}$ dont les microlentilles font $5.36\text{ }\mu\text{m}$ et recouvrent deux sous-pixels rectangulaires. Il propose un mode d'enregistrement dit "*dual-raw*" qui permet de sauvegarder les images de sous-ouverture droite et gauche dans le même fichier. Le logiciel DPRSplit[48] permet de convertir ce fichier en deux autres fichiers .dng séparés. En mesurant l'intensité des deux images selon l'angle d'incidence, on peut remonter aux réponses angulaires du capteur.

Nous avons d'abord créé une source collimatée étendue. Un trou micrométrique est placé devant une ampoule à tungstène, puis il est placé au foyer image d'un objectif photographique de focale 135 mm . L'objectif est diaphragmé pour réduire les aberrations géométriques. Le boîtier est fixé sur un support rotatif et on prend une photo tous les 3° , de -40° à 40° . Un exemple est donné sur la Figure 5.1. Plusieurs images *dark* (ici trois) sont prises avec le capuchon du boîtier couvrant le capteur, puis sont moyennées pour diminuer le bruit temporel. L'image moyenne est soustraite de toutes les autres photos, pixel à pixel. Le but[23] est de diminuer le bruit dû au courant d'obscurité, le bruit à motif fixe, et d'éliminer les pixels "chauds" (pixels défectueux). On extrait, dans la mesure du possible,

une zone de 300x300 pixels dont on prend la moyenne, ce qui donne un doublet d'intensité pour chaque angle d'incidence. A partir de 30° environ la monture du reflex produit une ombre sur le capteur et cela empêche d'extraire une zone suffisamment grande. La réponse totale est représentée sur la Figure 5.2 et les parties en-dessous de -30° et au-dessus de 30° sont sujettes à caution à cause de ce problème. J'ai normalisé les mesures d'un facteur variant avec le cosinus de l'angle pour corriger l'effet de la loi de Lambert.

La forme générale est celle attendue, mais on ne peut pas directement la comparer aux simulations de Kobayashi (Figure 2.13) car la taille de pixel est différente : $6.4\text{ }\mu\text{m}$ pour les simulations, $5.36\text{ }\mu\text{m}$ pour le 5D mk IV. On note toutefois que le seuil de 0.2 est à peine atteint à cause de "plateaux d'intensité" formé par le sous-pixel opposé. Ces plateaux sont inexpliqués car s'ils étaient dus à un offset de l'électronique de lecture, ils auraient du être corrigés par le dark. De plus au delà de -40° ou 40° , l'ombre de la monture masque totalement le capteur et l'intensité des deux sous-pixels est bien nulle donc cela exclut la lumière ambiante résiduelle.

Comme un pixel de $5.36\text{ }\mu\text{m}$ reste relativement grand, on peut émettre l'hypothèse que les pixels simulés par Canon sont plus aboutis que les pixels utilisés dans le 5D et limitent les fuites de lumière vers le sous-pixel opposé.

Cette expérience a aussi pour but de démontrer que l'on peut assez facilement mesurer les réponses angulaires d'un capteur plénoptique à l'aide d'un montage simple.

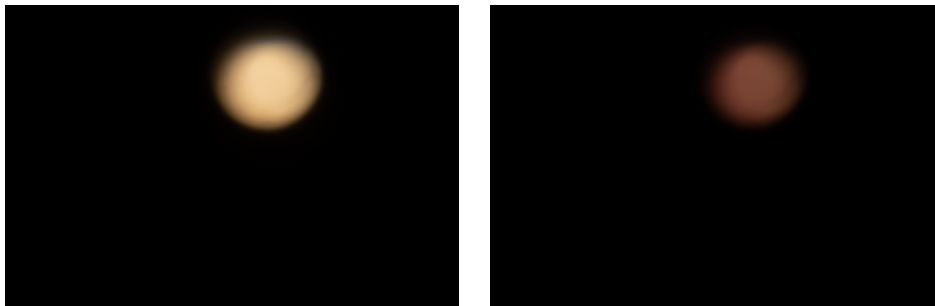


Figure 5.1. Les deux images de sous-ouverture d'une photo de la source collimatée.

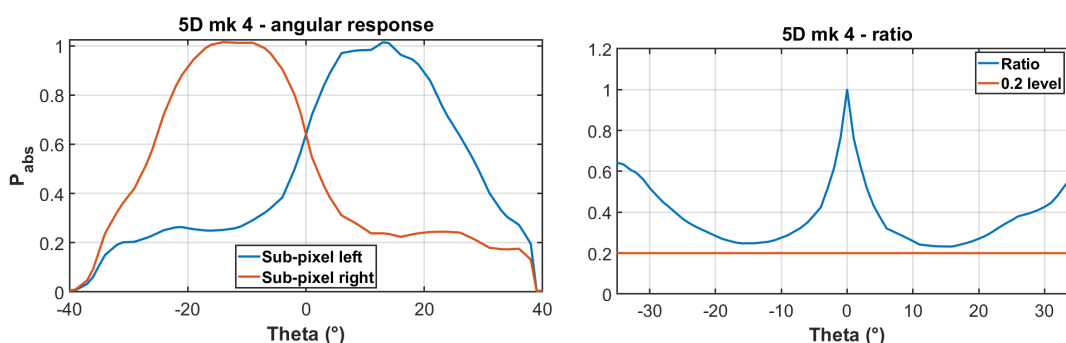


Figure 5.2. Réponse angulaire d'un Canon 5D mk IV. Les figures a) et b) présentent respectivement la puissance reçue par les deux sous-pixels, normalisée à 1, et le ratio associé.

5.2 Prototype de capteur quad-pixel

Dans cette partie je présente le prototype réalisé dans le cadre de ma thèse, je décris le banc de mesure et la façon dont les données ont été traitées, et enfin j'explique les principales sources d'incertitudes. Les résultats se concentrent sur les réponses angulaires et leur comparaison avec les simulations, mais je montre aussi des images prises avec le capteur.

5.2.1 Capteur

Notre prototype est basé sur un capteur fabriqué par STmicroelectronics (IMG140[73]) qui a des pixels de $1.75\ \mu\text{m}$, illuminés par la face arrière (BSI) et comportant des tranchées d'isolation (DTI). Il possède 788×1056 pixels au total, divisés en cinq bandes de sensibilités légèrement différentes. Ces différences sont dues au fait que la puce est au départ prévue pour comparer cinq circuits de lecture différents. Certaines colonnes à gauche et à droite sont recouvertes par un métal et ne sont pas utilisables. On trouve aussi des pixels désactivés entre les bandes pour pouvoir les séparer. La zone fonctionnelle a une taille de 768×940 pixels. Un schéma est présenté dans la Figure 5.3 a).

Le processus de fabrication a été arrêté par ST à la fin du processus *Back Side Illumination*[21], puis le CEA a finalisé la fabrication en gravant les microlentilles de $3.5\ \mu\text{m}$ au-dessus de groupes de 2×2 pixels et en ouvrant les plots de connexion. Le processus de fabrication des microlentilles est décrit dans la section suivante. Ce qu'il faut retenir ici, c'est qu'elles ont une base carrée, ce qui permet de recouvrir toute la surface des pixels. Il n'y a pas de filtre coloré entre la microlentille et le silicium. Plusieurs puces ont été produites et les 15 comportant le moins de défauts de fabrication ont été réservées pour les caractérisations.

La surface utile est découpée en 32 zones indiquées dans la Figure 5.3 b) :

- Une zone d'imagerie de 294×470 microlentilles centrées, s'étendant sur les bandes 2 à 5. Une image "flat" est nécessaire pour homogénéiser les différentes sensibilités des 4 bandes.
- Une bande de 21×470 microlentilles où le CRA est corrigé linéairement dans une direction. Les microlentilles sont centrées au début de la bande puis progressivement translatées jusqu'à atteindre un décalage de $1.467\ \mu\text{m}$, ce qui correspond à un CRA de 30° .
- 30 zones de 23×47 microlentilles qui sont décalées sur les axes x et y pour 30 valeurs de CRA différentes. Ces zones sont décrites dans le chapitre 3 et les valeurs des translations sont dans le Tableau 2.2. Le Tableau 5.1 résume les décalages. Ces zones sont toutes dans la bande 1 pour que la sensibilité des pixels soit la même entre chaque zone.

Chacune des 32 zones possède une couronne d'un pixel de large qui n'est pas recouverte par quoi que ce soit. Cela laisse de l'espace pour pouvoir décaler les microlentilles si nécessaire. Concernant le décalage linéaire en CRA, le coefficient à appliquer est idéalement de $1,467/470 = 0.0031213\ \mu\text{m}$ par microlentille. Cependant, pour faciliter le design du masque nous avons choisi de travailler avec des microlentilles de taille constante et légèrement inférieure à $3.5\ \mu\text{m}$. La taille idéale est de $(3,5 - 1,467)/470 = 3.49687\ \mu\text{m}$. L'arrondi au nanomètre inférieur produit un décalage en bout de ligne de $(3.5 - 3.496) * 470 = 1.88\ \mu\text{m}$, soit un CRA de 38° . Un arrondi au nanomètre supérieur produit un décalage en

bout de ligne de $(3.5 - 3.497) * 470 = 1.41 \mu\text{m}$, soit un CRA de 28° . C'est cette dernière option qui a été retenue car c'est elle qui s'approche le plus de la correction désirée. De plus mes simulations s'arrêtaient à un CRA de 30° : je n'aurais pas pu comparer le prototype pour des valeurs supérieures de toute façon. A noter que pour les microlentilles centrées ou pour les zones de CRA constant, les microlentilles font exactement $3.5 \mu\text{m}$.

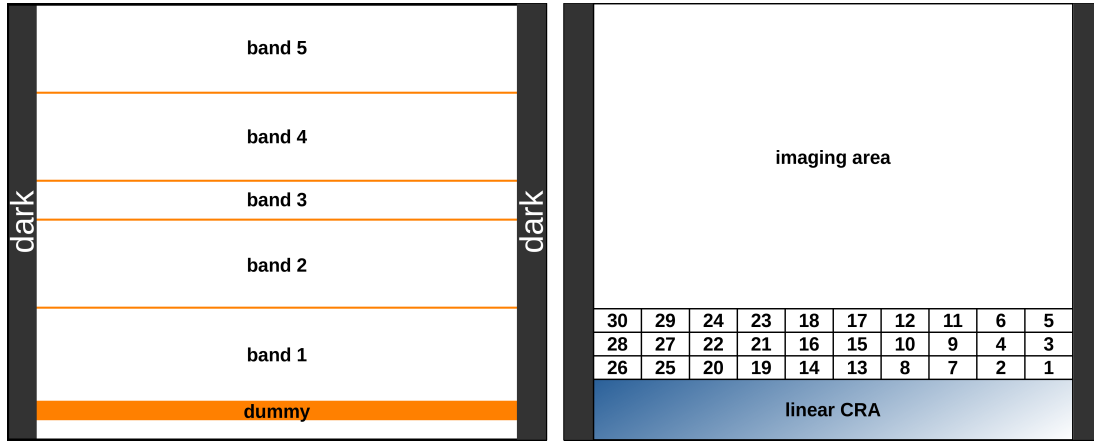


Figure 5.3. La figure a) montre la répartition des pixels sur la puce, les bandes 1 à 5 ont des sensibilités différentes et sont séparées par des colonnes désactivées. La figure b) montre les différentes zones de test.

Zone	φ ($^\circ$)	θ ($^\circ$)	Δ_x, Δ_y (μm)
1 à 6	0	5 à 30 par pas de 5	$\Delta_x = 0.237$ à 1.467 $\Delta_y = 0$
7 à 12	11.25	5 à 30 par pas de 5	$\Delta_x = 0.232$ à 1.439 $\Delta_y = 0.044$ à 0.292
13 à 18	22.5	5 à 30 par pas de 5	$\Delta_x = 0.219$ à 1.356 $\Delta_y = 0.086$ à 0.558
19 à 24	33.75	5 à 30 par pas de 5	$\Delta_x = 0.196$ à 1.228 $\Delta_y = 0.126$ à 0.821
25 à 30	45	5 à 30 par pas de 5	$\Delta_x = 0.165$ à 1.035 $\Delta_y = 0.165$ à 1.035

Tableau 5.1. Résumé des décalages des microlentilles en fonction du CRA.

5.2.2 Processus de fabrication

Les microlentilles sont réalisées en deux temps : on commence par fabriquer des microlentilles par lithographie et fluage thermique, puis on transfère leur forme dans un matériau placé en dessous au préalable. Cela permet de créer des microlentilles jointives et donc d'augmenter la surface de collection de la lumière. Ce processus de fabrication existant au CEA-Leti a été adapté pour la taille souhaitée ($3.5 \mu\text{m}$). De plus, le Leti avait aussi la charge d'ouvrir les plots de connexion (pads) qui permettent de relier électriquement la puce en silicium à son boîtier de lecture. La figure Figure 5.4 montre les grandes étapes de ce processus.

Les étapes nécessaires à la formation des microlentilles sont les suivantes :

- La résine dans laquelle sont formées les microlentilles est déposée par enduction centrifuge (spin coating) puis recuite (couche violette). Elle doit être gravée pour former les microlentilles.
- On dépose par spin coating la résine (couche rose) qui permet la formation d'une première couche de microlentilles. Un scanner procède à la lithographie avec un espacement compris entre 350 nm et 400 nm entre les motifs.
- Les microlentilles sont formées par fluage thermique, elles sont écartées d'environ 350 nm les unes des autres.
- Leur forme est transférée dans le premier matériau par gravure sèche (bombardement d'ions). Cette technique permet de supprimer l'espace entre les microlentilles.

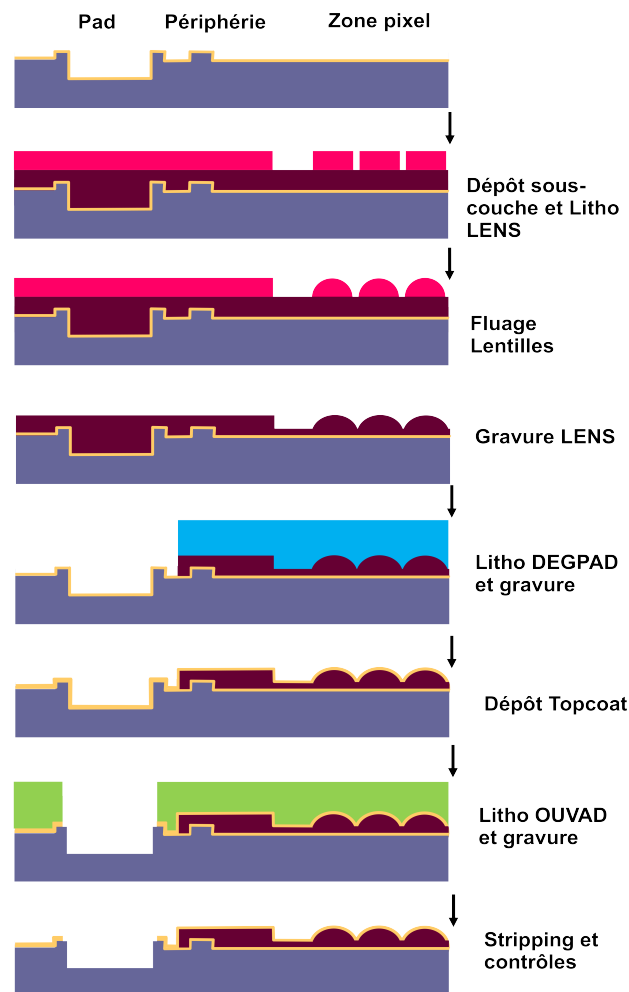


Figure 5.4. Étapes de fabrication des microlentilles.

Les étapes nécessaires à l'ouverture des pads sont les suivantes :

- On dépose une résine (couche bleue) qui permet ensuite de supprimer la résine microlentille au niveau des pads. Elle est retirée une fois les pads dégagés.
- On dépose une surcouche (topcoat) en oxyde de silicium sur les microlentilles. Sa fonction est de protéger les microlentilles et de servir de traitement anti-reflet. Le dépôt est réalisé par *Atomic Layer Deposition* ce qui permet la création de couches

ultra-fines. Cette couche doit, là encore, être retirée des pads.

- On dépose une résine (couche verte) et on grave le topcoat jusqu'au métal (aluminium). Cette résine est ensuite retirée.
- On peut à présent relier la puce au boîtier en soudant des fils en or (wire bonding) sur les pads.
- Gravure des plots jusqu'à ouverture, c'est dire jusqu'au métal (Al) sur lequel va se faire la soudure des fils en or reliant le circuit aux bornes du boîtier.
- retrait de la résine de lithographie (stripping)

La Figure 5.5 montre des clichés pris par un microscope électronique à balayage d'une coupe du prototype. On y voit les microlentilles en haut, le silicium au centre avec les DTI, et les lignes métalliques du circuit de lecture en bas. Les microlentilles sont bien jointives et leur uniformité en forme est bonne. Cependant on remarque qu'elles sont trop proches du silicium : la distance est d'environ 200 nm. Lors de mes simulations, la hauteur de l'empilement des couches valait $1.9\text{ }\mu\text{m}$. Ici le filtre coloré n'a pas pu être remplacé par une résine transparente, ce qui explique la hauteur manquante. Cela va dégrader la convergence de la microlentille et modifier les réponses angulaires du quad-pixel. Ce point est abordé en détail dans la partie résultat. De plus, les microlentilles ne sont peut être pas aussi bombées que dans les simulations, ce qui peut là encore dégrader la convergence et donc les réponses angulaires.

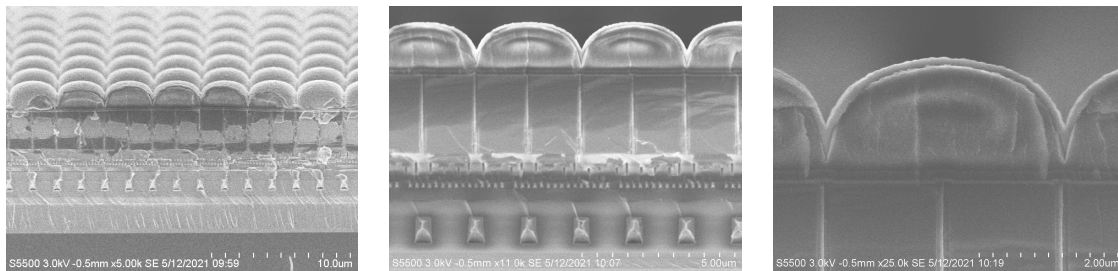


Figure 5.5. Coupe transverse du prototype vue par un microscope électronique à balayage.

5.2.3 Banc de mesure en angle

On utilise le banc optique montré sur la Figure 5.6 pour mesurer les réponses angulaires du prototype. La carte électronique permet de changer rapidement de puce. La source est une diode verte (532 nm) non collimatée placée à 40 cm du capteur. Les axes de rotation sont alignés sur le centre du capteur et sont pilotés par un logiciel codé en Lab-view. Pour les deux axes, les angles entre -15° et 15° sont échantillonnés par pas de 0.5° . Ceux de -50° à -15° et de 15° à 50° sont échantillonnés par pas de 2° . Cela fournit au total 12544 points de mesure. Des images "dark" sont prises à la fin des boucles de rotation : dix acquisitions en l'absence de source lumineuse et à même température ambiante. On les utilise ensuite de la même manière que dans l'expérience avec le Canon 5D mk IV : on en fait la moyenne (*master dark*) qui est soustraite aux autres images pixel à pixel.

La puissance de la LED est choisie pour maximiser le contraste angulaire à un angle d'incidence donné. Avec une puissance trop faible la mesure peut être dégradée par le bruit de lecture des pixels. A l'inverse, une source trop forte fera saturer les pixels et dégradera aussi la mesure. J'ai fixé un angle de 50° sur l'axe vertical, puis j'ai relevé le rapport entre

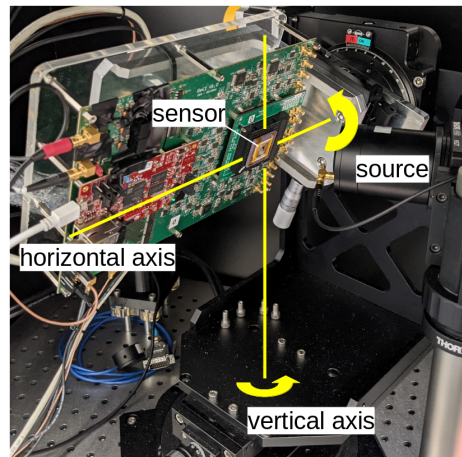


Figure 5.6. Banc de mesure en angle permettant des rotations selon les axes x et y .

les pixels de gauche et de droite en fonction du courant alimentant la LED (Figure 5.7). On détermine alors que l'intensité optimale pour la LED est de 20mA.

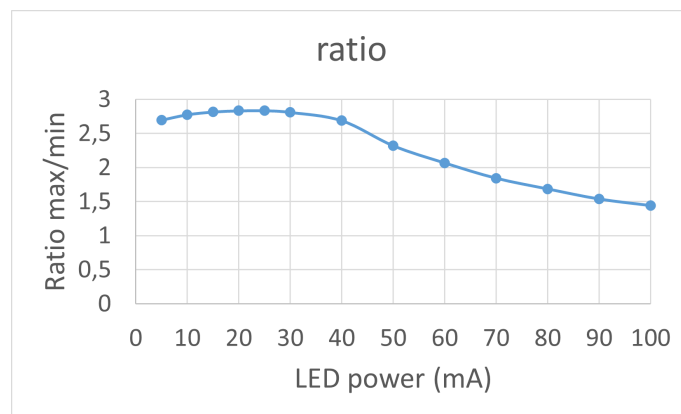


Figure 5.7. Courbe d'optimisation de la puissance de la LED

5.2.4 Traitement des données

Le capteur utilise la technique du double échantillonnage corrélé [84] pour diminuer le bruit à motif fixe (**Fixed Pattern Noise**). Les données sont donc enregistrées dans un vecteur ligne de $2 \times 788 \times 1056 = 1664256$ éléments qui contient pour chaque pixel la tension de référence et la tension après intégration. La première étape est de soustraire ces valeurs deux à deux afin d'extraire le signal utile proportionnel à l'intensité lumineuse reçue. On obtient un vecteur ligne de 788×1056 éléments. La seconde étape est de remettre l'image en deux dimensions pour pouvoir la visualiser. Comme la puce fournie par ST microelectronics a pour but de tester divers type de pixels, le circuit de lecture et son système d'adressage sont un peu particuliers. La façon de lire les pixels n'est pas standard pour un imageur et une remise en ordre est nécessaire. L'image est correctement affichée si l'on suit le Tableau 5.2, dans lequel les valeurs représentent les indices des éléments du vecteur ligne calculé précédemment.

0	3	4	...	2107	2108	2111
2112	2115	2116	...	4219	4220	4223
1	2	5	...	2106	2109	2110
2113	2114	2117	...	4218	4221	4222
...						
...						832125

Tableau 5.2. Remise en ordre des pixels pour passer du vecteur ligne à une matrice image en deux dimensions, où chaque nombre est l'indice d'un élément du vecteur.

Une fois l'image obtenue, on soustrait le dark puis on en extrait les SAIs. En calculant la moyenne de chaque zone par SAI pour tous les points de mesure, on obtient finalement les réponses angulaires par zone et par SAI. Ces réponses sont mesurées sur une base de rotation sur les axes x et y . Les angles associés sont appelés \widehat{H} et \widehat{V} . Pour pouvoir les comparer facilement aux simulations, il faut changer de base et passer sur des coordonnées (φ, θ) .

On considère un vecteur unité \vec{u} perpendiculaire au plan du capteur lorsque $\widehat{H} = 0$ et $\widehat{V} = 0$. Une rotation sur l'axe vertical projette \vec{u} sur le capteur selon x_0 dans le plan du capteur et selon z_0 perpendiculairement au capteur. La rotation sur l'axe horizontal projette z_0 selon y_1 et z_1 , respectivement dans le plan du capteur et perpendiculairement au capteur. On a les relations suivantes :

$$z_0 = \sin(90 - \widehat{V}) \quad (5.1)$$

$$x_0 = \cos(90 - \widehat{V}) \quad (5.2)$$

$$y_1 = z_0 \sin(\widehat{H}) \quad (5.3)$$

$$z_1 = z_0 \cos(\widehat{H}) \quad (5.4)$$

r est la projection finale de \vec{u} dans le plan du capteur après les rotations effectuées. On obtient φ et θ en utilisant les relations classiques de passage entre coordonnées cartésiennes et polaires :

$$r = \sqrt{x_0^2 + y_1^2} \quad (5.5)$$

$$\theta = 90 - \arctan(z_1/r) \quad (5.6)$$

$$\varphi = \begin{cases} \arctan(y_1/x_0) & , \text{ si } x_0 > 0 \text{ et } y_1 \geq 0 \\ \arctan(y_1/x_0) + 360 & , \text{ si } x_0 > 0 \text{ et } y_1 < 0 \\ \arctan(y_1/x_0) + 180 & , \text{ si } x_0 < 0 \\ 90 & , \text{ si } x_0 = 0 \text{ et } y_1 > 0 \\ 270 & , \text{ si } x_0 = 0 \text{ et } y_1 < 0 \\ 0 & , \text{ else} \end{cases} \quad (5.7)$$

L'échantillonnage régulier en \widehat{H} et \widehat{V} produit un nuage de points non régulier en (φ, θ) . Pour une meilleure visualisation, j'ai décidé d'interpoler les données sur une grille régulière et plus fine en (φ, θ) . φ va de 0° à 360° et θ de 0° à 65° par pas de 1° . Les données en \widehat{H} et \widehat{V} sont d'abord placées au plus proche voisin dans la grille (φ, θ) à l'aide des formules précédentes. La fonction `griddata` de Scipy en Python est ensuite utilisée pour trouver

les points manquants à l'aide d'une interpolation linéaire. Comme l'intensité des pixels correspond aux tensions numérisées par les convertisseurs analogiques-numériques, j'ai décidé de les normaliser à 1 par zone dans les figures qui vont suivre. Les données n'ont pas été altérées au delà des manipulations décrites ci-dessus, en particulier on ne corrige pas les sources d'incertitudes explicitées dans le paragraphe suivant.

La zone de CRA linéaire est particulière. Dans ce cas-là, je calcule le ratio gauche/droite par microlentille et je calcule la moyenne par colonnes de 21 microlentilles. Comme on s'intéresse à un seul axe, on a directement $\widehat{V} = \theta$. Le ratio est tracé en fonction de la position sur le capteur.

5.2.5 Facteurs d'erreurs

Les incertitudes concernent l'angle d'incidence de mesure et l'intensité lumineuse reçue. Concernant l'angle d'incidence, les sources d'erreurs sont les suivantes :

- le mauvais alignement de la source face à l'axe du capteur.
- le montage de la carte électronique sur la platine.
- le montage du capteur sur la carte électronique.
- le placement des microlentilles par rapport aux pixels.
- la diode source non collimatée
- le mauvais alignement des axes de rotation sur le capteur

Les quatre premières erreurs citées produisent un décalage global que l'on peut caractériser en mesurant les ratios gauche/droite et haut/bas sur les zones où les microlentilles sont centrées. La coordonnée où le pic central vaut 1 donne la valeur du décalage. D'après nos mesures, le décalage maximum relevé est de $\pm 4^\circ$.

Le fait que la diode ne soit pas collimatée fait que l'angle d'incidence effectif varie entre le centre et les bords du capteur. Ce phénomène est amplifié si les axes de rotation ne sont pas exactement dans le plan du capteur. La Figure 5.8 présente ces deux erreurs sur le même schéma.

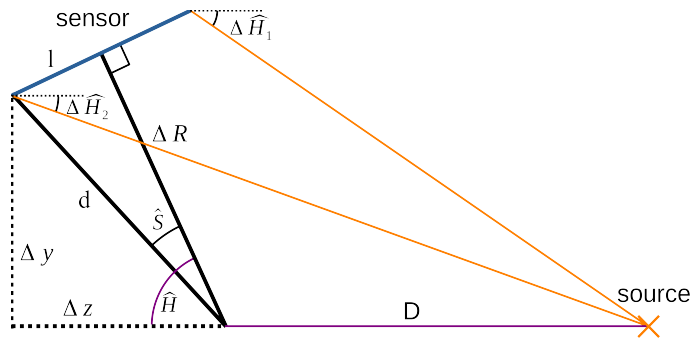


Figure 5.8. Schéma des incertitudes en angles d'incidence. l est la moitié de la taille du capteur, D est la distance entre la source et l'axe de rotation, \widehat{H} est l'angle de rotation et ΔR est l'erreur d'alignement entre l'axe de rotation et le capteur. $\Delta \widehat{H}_1$ et $\Delta \widehat{H}_2$ sont les erreurs sur l'angle d'incidence maximum mesuré aux bords du capteur.

Sur la Figure 5.8, d et \widehat{S} s'écrivent :

$$d = \sqrt{l^2 + \Delta R^2} \quad (5.8)$$

$$\widehat{S} = \text{asin} \left(\frac{l}{d} \right) = \text{asin} \left(\frac{l}{\sqrt{l^2 + \Delta R^2}} \right) \quad (5.9)$$

Selon le bord du capteur que l'on considère et selon le signe de ΔR , on doit ajouter ou soustraire \widehat{S} à \widehat{H} . Sur la figure, $\Delta\widehat{H}_1$ est lié à $\widehat{H} + \widehat{S}$ et $\Delta\widehat{H}_2$ est lié à $\widehat{H} - \widehat{S}$. Δy and Δz valent :

$$\Delta y = d \sin(\widehat{H} - \widehat{S}) \quad (5.10)$$

$$\Delta z = d \cos(\widehat{H} - \widehat{S}) \quad (5.11)$$

L'erreur maximale en angle d'incidence sur les bords du capteur vaut donc :

$$\Delta\widehat{H}_2 = \text{atan} \left(\frac{\Delta y}{D + \Delta z} \right) \quad (5.12)$$

En combinant les équations on obtient au final :

$$\Delta\widehat{H}_{1,2} = \text{atan} \left(\frac{\sqrt{l^2 + \Delta R^2} \sin \left(\widehat{H} \pm \text{asin} \left(\frac{l}{\sqrt{l^2 + \Delta R^2}} \right) \right)}{D + \sqrt{l^2 + \Delta R^2} \cos \left(\widehat{H} \pm \text{asin} \left(\frac{l}{\sqrt{l^2 + \Delta R^2}} \right) \right)} \right) \quad (5.13)$$

Enfin, le changement de coordonnées expliqué dans le paragraphe Section 5.2.4 peut éventuellement produire une erreur de $\pm 0.5^\circ$ car la grille d'arrivée a un pas de 1° .

L'intensité lumineuse étant proportionnelle au carré de la distance, l'erreur relative causée par ΔR est donnée par $(D + \Delta R)^2/D^2$. Or le banc de mesure a un décalage de $400 \mu\text{m}$ entre son axe de rotation vertical et son axe de rotation horizontal. Lorsque l'un des deux axes est aligné sur le capteur, l'autre est forcément devant ou derrière le plan du capteur.

Le Tableau 5.3 résume les valeurs utiles pour calculer les incertitudes. Une application numérique donne une variation relative de l'intensité de $\pm 0.2\%$ et une variation de l'angle d'incidence de $\pm 0.14^\circ$.

Taille du capteur	1.848 mm
Distance de la source	40 cm
Distance entre l'axe horizontal et l'axe vertical	400 μm

Tableau 5.3. Valeurs des paramètres permettant de calculer les incertitudes en angle.

5.2.6 Réponses angulaires

Dans cette partie je compare les mesures aux simulations. Les quatre SAIs A, B, C et D sont toujours respectivement en haut à gauche, en haut à droite, en bas à gauche et en bas à droite.

5.2.6.1 Microlentilles centrées

La Figure 5.9 montre une mesure dans une seule direction ($\varphi = 0$) des pixels de gauche et de droite ainsi que le ratio associé. On rappelle aussi les courbes issues des simulations. Bien que le ratio du prototype n'atteigne jamais 0.2, le contraste angulaire dépasse 1 après $\theta = 25^\circ$: il vaut 1.35 à 25° exactement.

La Figure 5.10 rappelle les réponses angulaires simulées en deux dimensions et la Figure 5.11 présente les réponses des quatre bandes du prototype où les microlentilles sont

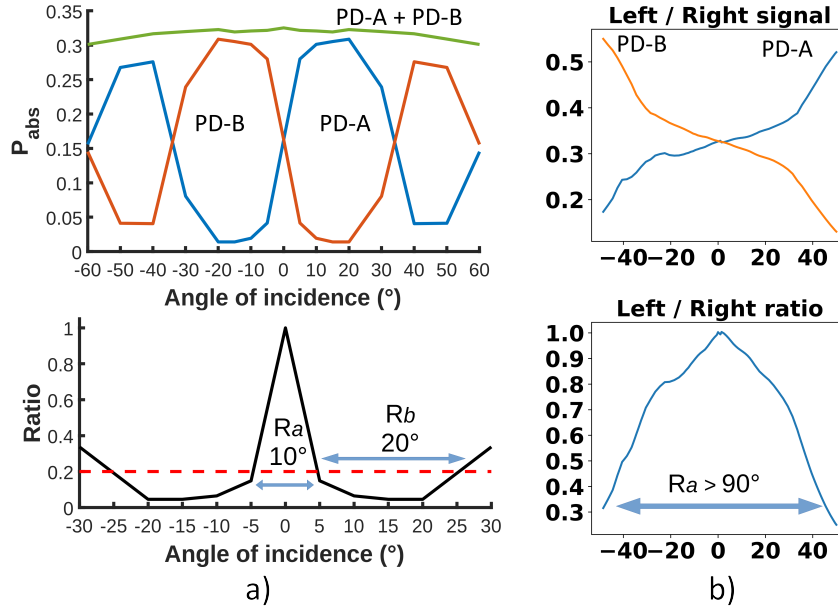


Figure 5.9. La figure a) montre les réponses de gauche et de droite d'un quad simulé puis son ratio associé en-dessous. La figure b) montre les courbes associées à la mesure sur le prototype.

centrées. En simulation, les SAIs voient bien un quart de la pupille de l'objectif principal, aux facteurs de crosstalk près. Par exemple pour la SAI-A et φ compris entre 270° et 360° , la zone utile s'étend d'environ $\theta = 5^{\circ}$ jusqu'à environ $\theta = 45^{\circ}$ dans la diagonale. Ce n'est pas le cas pour le prototype où les réponses sont quasiment similaires indépendamment de l'angle jusqu'à au moins 25° . L'effet plénoptique devient donc visible à partir de $\theta = 25^{\circ}$ quel que soit φ , bien que le contraste angulaire soit le plus marqué autour de 40° . Cela signifie que les SAIs vont être identiques et donc que leur exploitation pour une mesure d'autofocus va être très difficile voire impossible, à moins d'utiliser un objectif très ouvert. Si on se base sur la valeur de 25° trouvée avec la mesure sur une dimension, cela signifie qu'il faut utiliser une ouverture plus grande que F#1.1.

Ceci étant dit, le prototype présente tout de même des réponses asymétriques par quarts et les sous-pixels répondent dans la plage angulaire attendue (au moins en φ), là où un pixel classique présenterait des réponses totalement centrées. La tendance globale des courbes est donc la bonne.

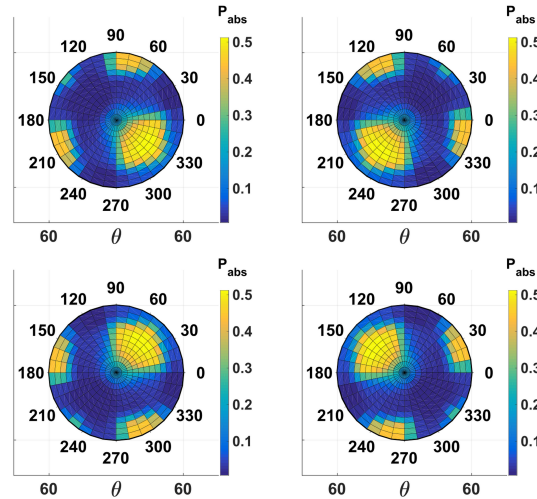


Figure 5.10. *Rappels des réponses angulaires des quatre sous-pixels en simulation.*

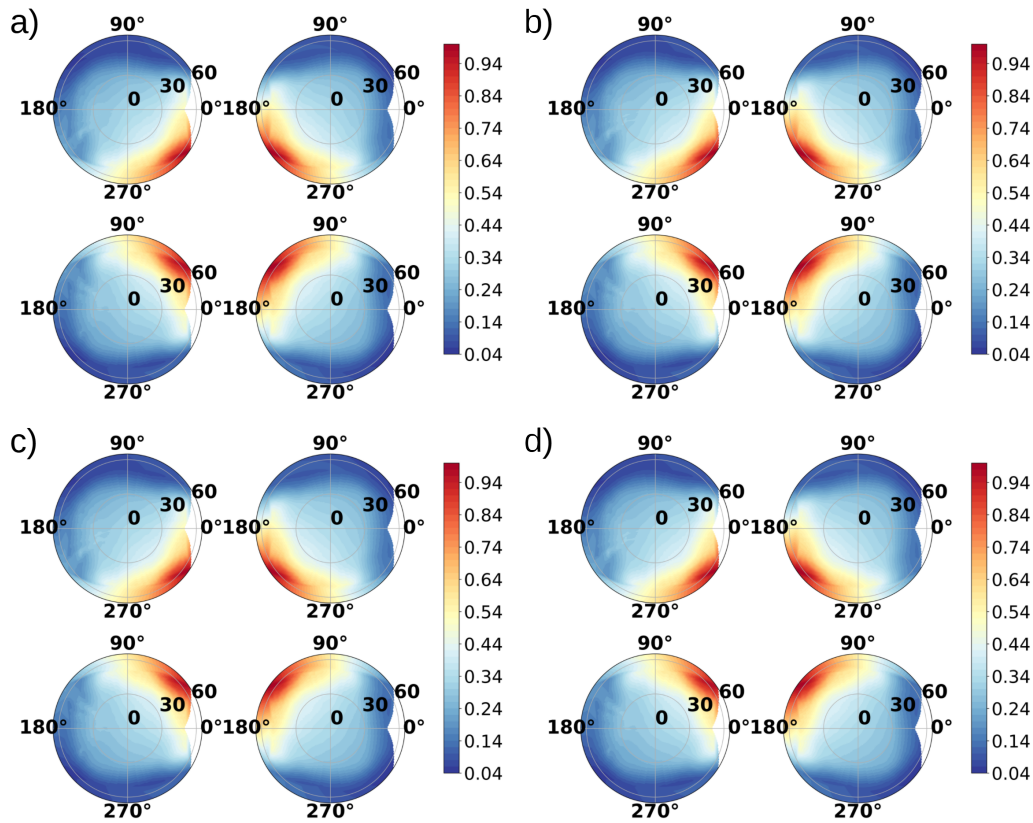


Figure 5.11. *Réponses angulaires des quatre bandes où les microlentilles sont centrées.*

5.2.6.2 Zones de Chief Ray Angle constant

Je fournis ici toutes les mesures des 30 zones de CRA constants. On s'attend normalement à ce que les puissances absorbées soient les mêmes entre les sous-pixels lorsque l'angle d'incidence vaut celui du CRA corrigé. Ce n'est pas le cas ici à cause de la mauvaise hauteur de la microlentille. Cependant leur décalage produit bien un effet : on constate

que les lobes des réponses angulaires sont translatés en fonction de φ_{cra} et θ_{cra} . De plus, ces résultats restent cohérents avec ce à quoi on pouvait s'attendre. Par exemple pour $\varphi_{cra} = 45$ et $\theta_{cra} = 30$ la microlentille est à l'aplomb du sous-pixel B, donc pour des angles d'incidence faibles, c'est ce pixel qui doit recevoir toute la puissance. C'est bien le cas, comme on peut le voir sur la Figure 5.16 d). La puissance reçue par les sous-pixels C et D provient des microlentilles adjacentes qui elles-aussi sont décalées quasiment au-dessus des sous-pixels voisins.

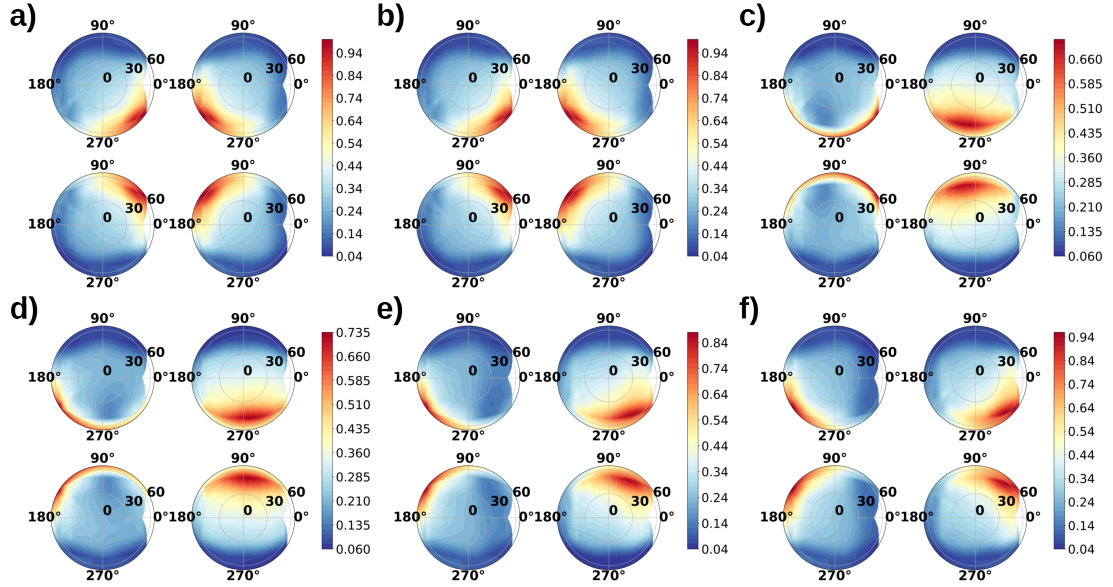


Figure 5.12. Réponses angulaires des zones où $\varphi_{cra} = 0^\circ$ et $\theta_{cra} = 0$ à 30° .

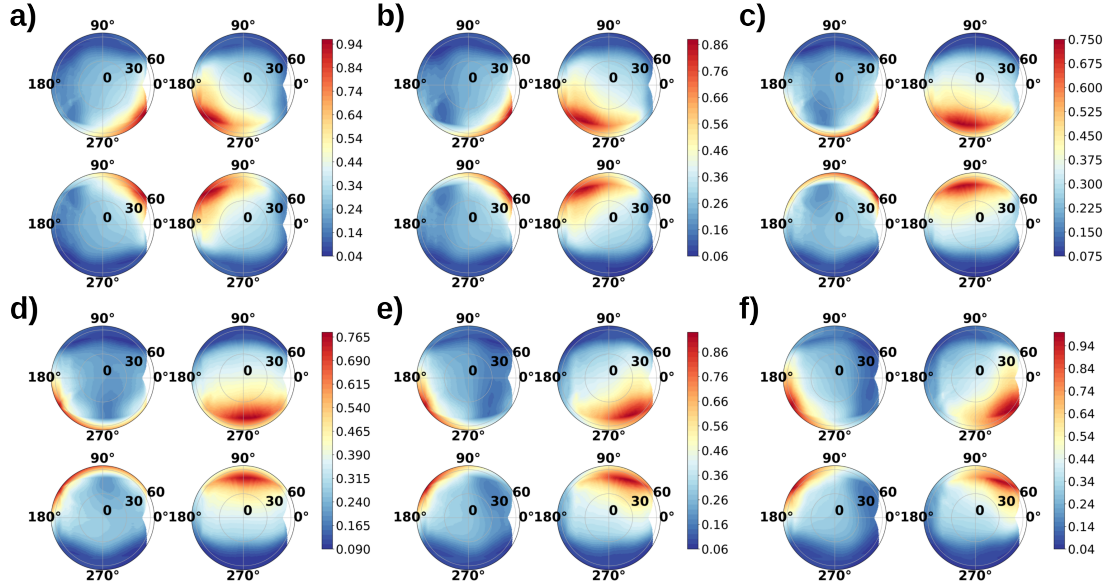


Figure 5.13. Réponses angulaires des zones où $\varphi_{cra} = 11.25^\circ$ et $\theta_{cra} = 0$ à 30° .

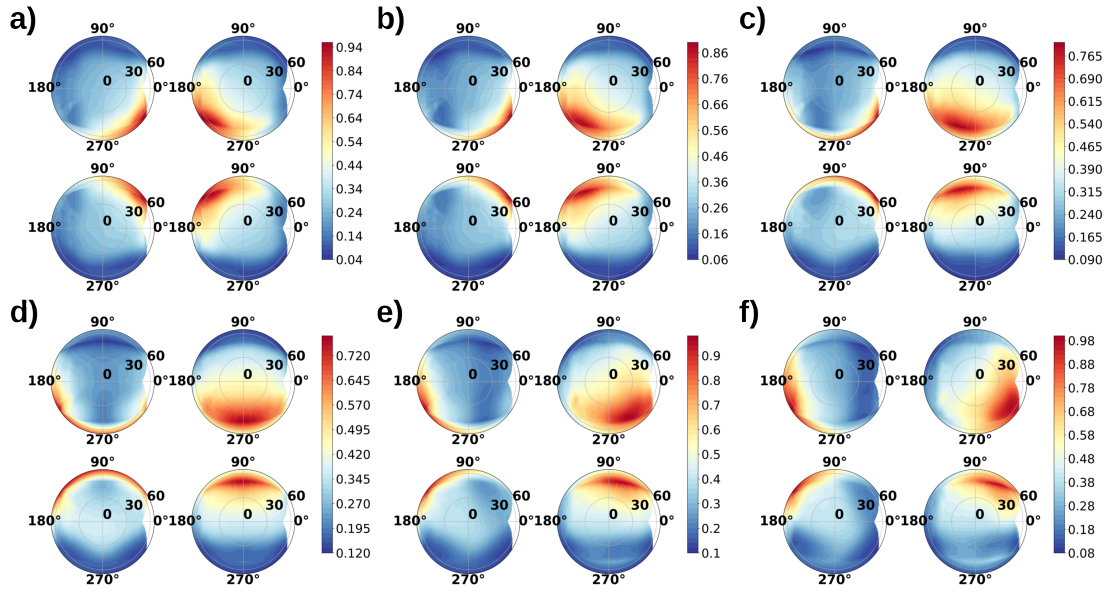


Figure 5.14. Réponses angulaires des zones où $\varphi_{cra} = 22.5^\circ$ et $\theta_{cra} = 0$ à 30° .

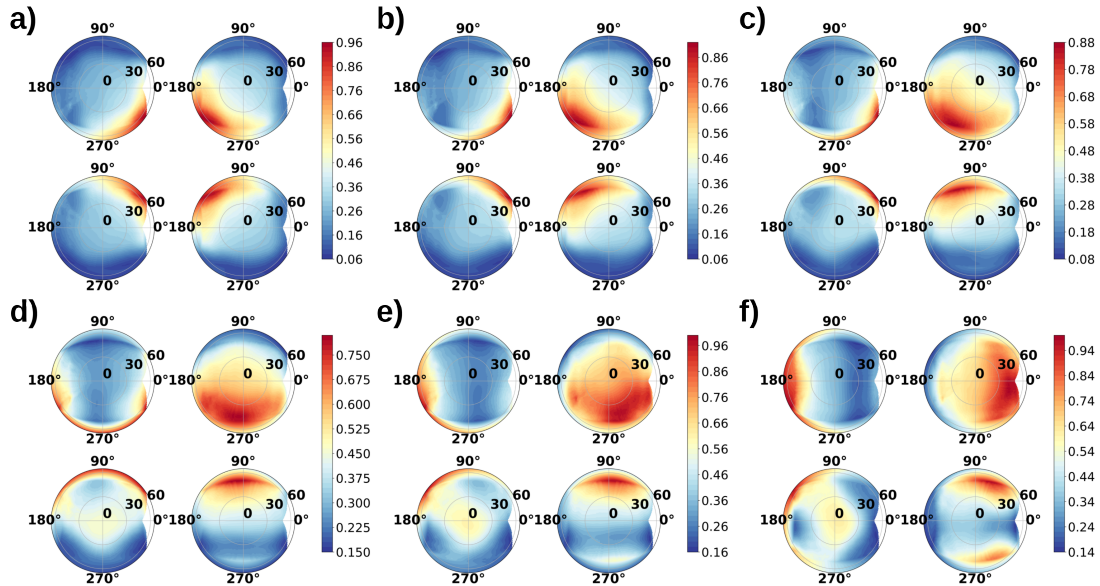


Figure 5.15. Réponses angulaires des zones où $\varphi_{cra} = 33.75^\circ$ et $\theta_{cra} = 0$ à 30° .

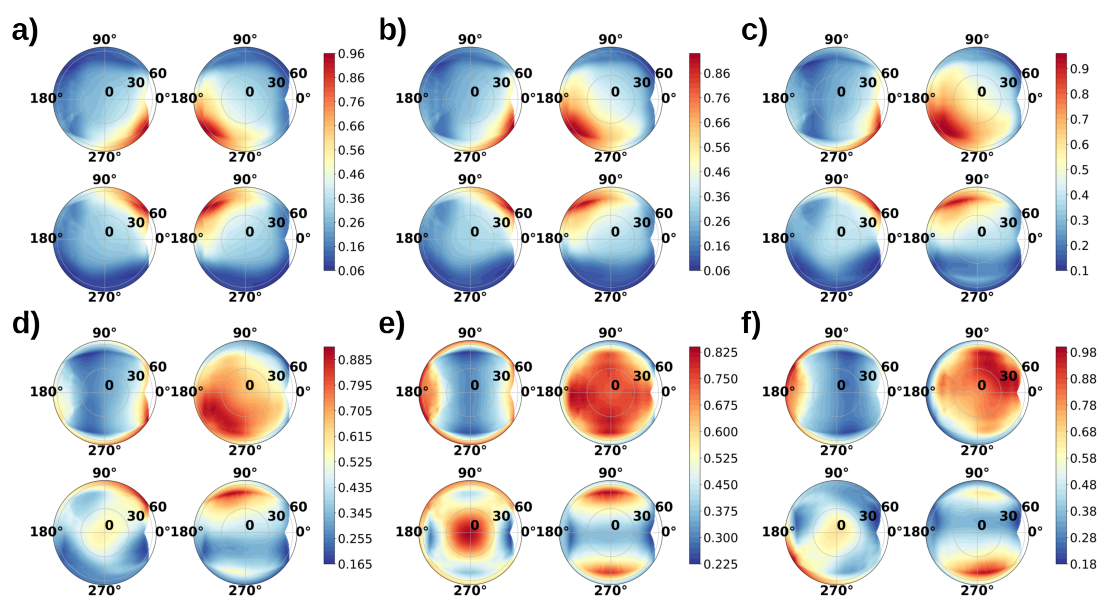


Figure 5.16. Réponses angulaires des zones où $\varphi_{cra} = 45^\circ$ et $\theta_{cra} = 0$ à 30° .

5.2.6.3 Zone de Chief Ray Angle linéaire

Je termine la caractérisation du capteur par la zone de CRA linéaire. La Figure 5.17 présente le ratio gauche/droite par microlentille sous plusieurs angles d'incidence, mesuré à gauche et attendu à droite. La microlentille située à droite de la bande (coordonnée 470 en abscisses) est centrée sur le groupe de 2x2 pixels tandis que celle située à gauche de la bande (coordonnée 0) est décalée de $1.41 \mu\text{m}$. On s'attend à ce que pour un angle d'incidence donné, le ratio soit linéaire et croise la valeur 1 à l'endroit où la microlentille est décalée exactement pour cet angle. Une autre façon de voir les choses est de considérer une seule microlentille : le ratio devrait augmenter avec l'angle d'incidence.

Comme les microlentilles n'ont pas assez d'espace pour converger correctement, le ratio mesuré est en fait toujours relativement proche de 1. Pour celles qui sont centrées, le crosstalk entre les sous-pixel est dominant. Pour celles qui sont décentrées, c'est le crosstalk entre microlentilles qui domine.

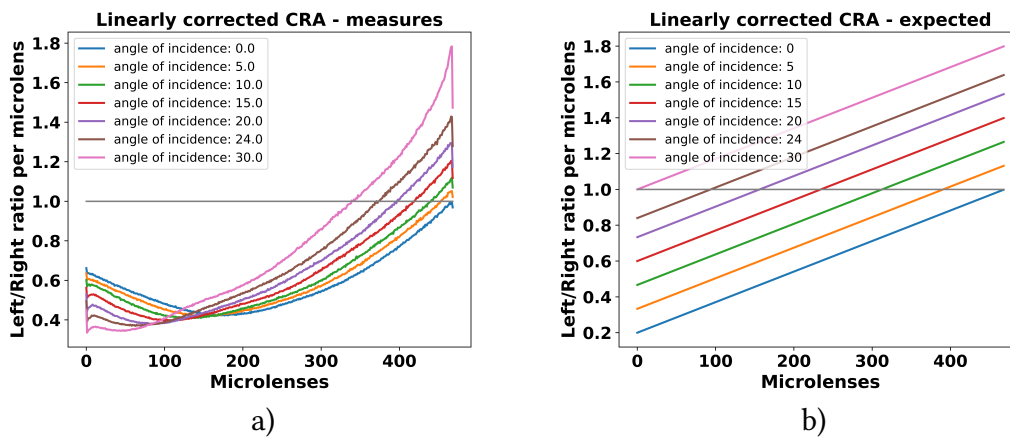


Figure 5.17. Ratio par microlentille mesuré et attendu.

5.2.7 Images

Dans cette dernière partie, je montre des images prises avec le prototype. Les premières sont des captures d'une mire de bruit blanc binaire (pixels valant 0 ou 1). Cela permet d'avoir une meilleure visualisation des disparités (si elles existent) car les motifs ont des contours nets.

Les images sont prises avec un objectif 17mm F#0.95. La mise au point est faite à 30 cm, la première image est dans le plan de netteté et la seconde 20 mm derrière. J'ai extrait une ligne et une colonne des signaux gauche/droite et haut/bas de la mire floue sur la Figure 5.19. On remarque un décalage d'un pixel entre les SAIS, surtout visible sur la visualisation de la ligne gauche/droite. Cependant en appliquant la formule Équation (1.5), on devrait trouver une tache de flou dont le rayon fait 0.11mm environ, soit environ 30 microlentilles. La disparité devrait donc faire au moins 30 pixels.

Pour essayer de mesurer des disparités plus nettes, nous avons utilisé un objectif 25mm F#0.95 puis essayé de masquer sa partie centrale. Le but est de couper la partie qui cause le crosstalk entre les sous-pixels et de conserver uniquement l'anneau externe qui devrait



Figure 5.18. Une partie de la mire de bruit utilisée.

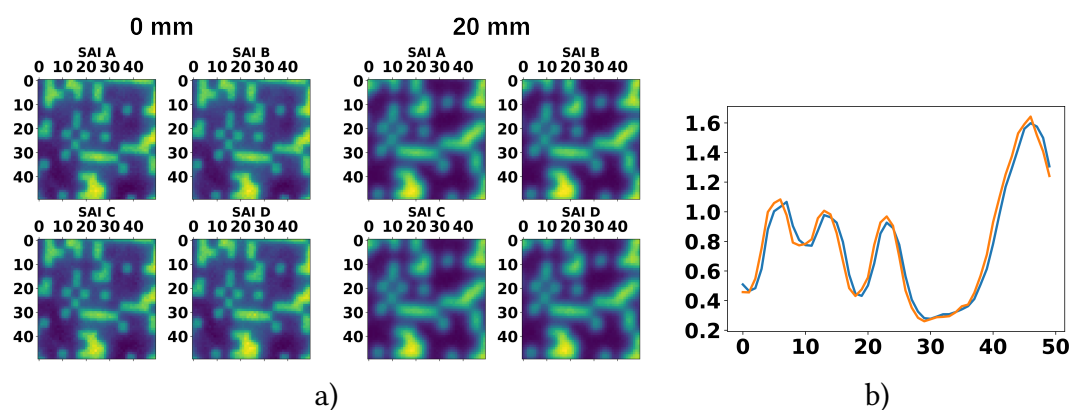


Figure 5.19. SAIs d'une mire de bruit au point et 20mm derrière le plan de netteté. Signaux gauche/droite et haut/bas extraits d'une ligne et d'une colonne.

amener le signal de disparité. Idéalement, le masque doit être placé dans la pupille de sortie de l'objectif mais cela nécessiterait un démontage complet. De plus, les images présentées ici sont en couleur car il s'agit de la fusion de 3 acquisitions avec des filtres colorés rouge, vert et bleu placés devant l'objectif.



Figure 5.20. Objectif photo Mitakon 25mm F#0.95 avec un masque devant la lentille frontale. Le masque a pour but d'améliorer le contraste angulaire.

Les images sont montrées sur les Figure 5.21 et Figure 5.22, respectivement sans et avec masque. La scène utilisée est composée d'un fût d'objectif au premier plan, d'une canette au second plan et enfin d'un carton avec du texte en arrière-plan. La mise au point est effectuée à 1 mètre sur la canette. Les signaux gauche/droite sont montrés sur les

graphiques à droite. Quelle que soit la configuration, je n'ai pas réussi à voir de disparités sur les images.

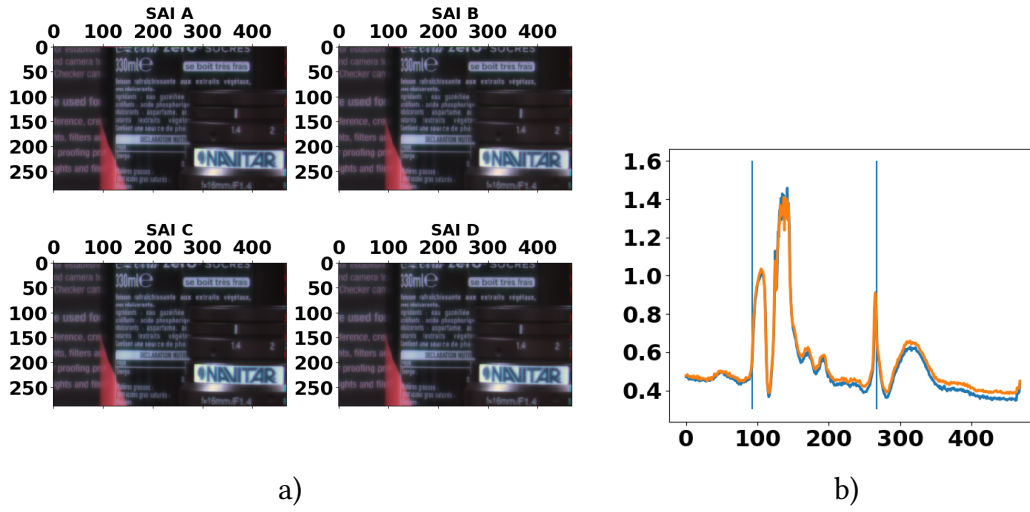


Figure 5.21. Scène canette sans le masque couvrant l'objectif. Une coupe horizontale est représentée à droite, sans montrer de disparités claires.

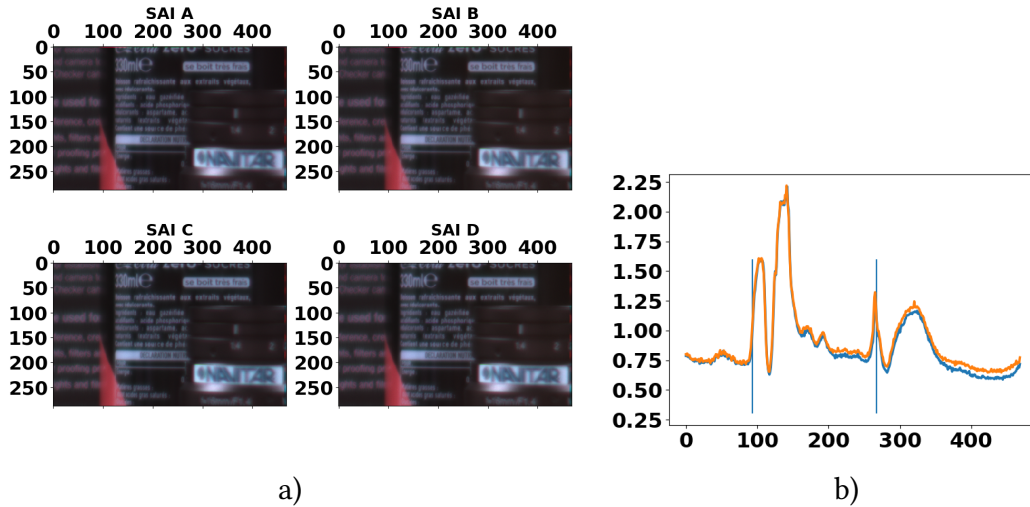


Figure 5.22. Scène canette avec le masque. La coupe horizontale ne présente pas de disparités claires ici non plus, malgré le masque qui devrait augmenter le contraste angulaire.

5.3 Conclusion

Après avoir validé le principe de la mesure sur un Canon 5D mk IV, nous avons défini, réalisé et enfin caractérisé un prototype de capteur quad-pixel à l'aide de STmicroelectronics et du CEA-Leti. Malgré la distance entre la microlentille et le silicium qui est trop élevée, nous avons pu mesurer les réponses angulaires puis les comparer aux simulations.

Les graphiques en deux dimensions montrent que la tendance est bonne : les sous-pixels répondent bien en fonction des plages angulaires attendues. Le problème principal est que cette réponse arrive pour des angles d'incidence trop élevés pour être utilisables : la dynamique en angle est très dégradée. Les images prises avec le prototype confirment malheureusement qu'il est très difficilement exploitable en tant que capteur plénoptique car cela nécessiterait une optique encore plus ouverte que $F\#0.95$ et dont la partie centrale serait masquée pour améliorer le contraste angulaire. Cela confirme par contre le besoin d'optimiser finement les pixels et surtout de réduire le crosstalk entre sous-pixels le plus possible. Un second démonstrateur devrait être réalisé l'année prochaine, probablement avec une résine de couleur verte pour venir combler la distance entre la microlentille et le silicium. Dans ce cas on pourrait comparer directement les réponses angulaires des simulations à celles réelles, mais on perdrait la possibilité de faire des images couleurs RVB.

Le travail de caractérisation est en cours de publication dans un article du journal Optics Letter (pas de référence disponible à l'heure où j'écris ces lignes).

Conclusion et perspectives

6.1 Résumé

6.1.1 Manuscrit

L'objectif de cette thèse est d'analyser conjointement l'architecture du pixel et les algorithmes de traitement d'images pour optimiser ces deux aspects de manière globale et au final d'en tirer le meilleur parti. J'ai d'abord mené des simulations complètes au niveau du pixel, avec des microlentilles centrées puis décentrées, ce qui a permis de comprendre les sources de crosstalk. Je propose plusieurs architectures de pixel moins conventionnelles pour tenter d'améliorer les performances du pixel lui-même. Ces simulations ont fourni des réponses angulaires en quatre dimensions que j'ai ensuite utilisées au sein d'une méthode de rendu pour pouvoir générer des images plus réalistes du point de vue du capteur. Cette méthode de rendu mixte fait écho à la formulation de formation des images de sous-ouverture développée en parallèle, basée sur la notion de pondération angulaire. Les images générées constituent une base d'images quad-pixel qui ont servi d'une part à étudier l'impact de la diffraction des microlentilles sur les algorithmes de traitement d'images, et d'autre part, à entraîner des réseaux de neurones sur des tâches comme le dématricage couleur, la correction de la diffraction et l'estimation de profondeur. J'ai montré que la diffraction était bien un facteur limitant en prenant pour exemple la correction des aberrations de l'objectif. Cela confirme le besoin d'optimiser finement les pixels au niveau de leur architecture. Les utilisations des réseaux de neurones se sont avérées prometteuses en particulier pour le dématricage et la correction de la diffraction. Les résultats sont moins bons concernant l'estimation de profondeur : notre base de données paraît trop réduite pour entraîner correctement un réseau sur cette tâche plus complexe. Enfin, j'ai présenté deux expériences visant à mesurer les réponses angulaires réelles, premièrement sur un capteur dual-pixel Canon via le boîtier 5D mk IV, et deuxièmement sur un prototype fabriqué par ST microelectronics et finalisé au CEA.

6.1.2 Publications

Le travail réalisé pendant cette thèse a donné lieu à plusieurs publications, dont deux articles pour l'*Optical society of America* et un article de conférence pour *SPIE*. Deux brevets ont aussi été déposés mais dont les détails ne peuvent pas encore être divulgués.

Liste des publications :

- *Simulations FDTD et génération d'images de synthèse pour la conception d'un capteur plénoptique 2x2 pixels*, Chataignier, Vandame, Vaillant, Présentation orale, GDR-ISIS, novembre 2018
- *Joint electromagnetic and ray-tracing simulations for quad-pixel sensor and computational imaging*, Chataignier, Vandame, Vaillant, article de journal, Optic Express, 2019
- *Deep learning applied to quad-pixel plenoptic images*, Chataignier, Vandame, Vaillant, article de conférence + présentation orale, Computational Optics - SPIE, 2021
- **Publication en cours.** *Angular response of a quad-pixel plenoptic sensor prototype*, Chataignier, Vandame, Vaillant, article de journal, Optic Letter, 2021 (**prévisionnel**)

Liste des brevets :

- Structures dans le pixel diminuant le crosstalk entre les microlentilles
- Positionnement optimal des filtres colorés dans le cas d'un capteur "nona-pixel" (3x3 sous-pixels)

6.2 Poursuites et ouvertures

Plusieurs points doivent être consolidés ou finalisés. En premier lieu, le prototype réalisé n'avait pas la bonne hauteur de résine, ce qui a fortement dégradé la convergence de la microlentille. Une deuxième version devrait voir le jour pour corriger ce problème, ce qui permettrait de vraiment comparer la réalité aux simulations effectuées. Le deuxième point important concerne l'estimation de profondeur. Il s'agirait soit de consolider le dataset d'images de synthèse avec des scènes plus variées en couleur, en texture et en profondeur, soit idéalement de se servir directement du prototype pour capturer de vraies images quad-pixel. On pourrait ainsi s'affranchir des contraintes du rendu : le temps de rendu principalement, mais aussi le bruit du lancé de rayon et le nombre de scènes différentes assez restreint pour la version 2 de PBRT. Enfin, il faudrait continuer d'optimiser les réseaux de neurones développés pour le dématricage couleur et la correction de la diffraction. Par exemple les rendre plus performants pour le même nombre de paramètres libres, ou plus légers à performance égale. Le développement de nouvelles fonctions de coût, spécifiques à ces deux problèmes, serait aussi à creuser.

Au delà du fait de poursuivre ce travail, il existe d'autres pistes de recherche plus ou moins éloignées de ce qui a été présenté ici. Un axe évident est la prise en compte de l'objectif utilisé pour former l'image sur le capteur. On peut imaginer une démarche de co-conception plus globale prenant à la fois en compte le pixel, l'optique et la chaîne de traitement d'images. Le modèle de formation des images de sous-ouverture développé dans le chapitre 4 pourrait nous y aider. Une solution potentielle serait de produire une PSF globale carrée via l'objectif. Le quad pixel partageant la PSF en quatre de manière symétrique, on obtiendrait des SAIs convoluées par des fonctions portes. Ces dernières, contrairement aux quarts de disque, seraient plus facilement reconnaissables pour l'algorithme de block matching classique.

On peut remarquer que le lien entre le pixel et les algorithmes s'est fait plutôt à sens unique. J'ai insisté sur le fait que la diffraction par la microlentille influait négativement sur les performances des algorithmes. Un réseau de neurones a d'ailleurs été dédié à la correction du crosstalk entre les SAIs. Cependant, les possibilités offertes par les algo-

rithmes n'ont pas donné lieu à des modifications directes sur les architectures du pixel. Les optimisations du pixel proposées découlaient plus du critère de performance utilisé que des possibilités offertes par le deep learning. Il serait envisageable par exemple de partir d'un réseau de neurones pré-existant, sachant reconnaître des réponses angulaires particulières, et d'adapter la structure du pixel pour qu'il ait ces réponses.

Une autre piste de recherche serait le remplacement des microlentilles réfractives par des structures diffractives planaires comme les Fresnel Zone Plate[70, 66] et les métasurfaces [85]. On pourrait tenter de définir une fonction optique qui produirait des réponses angulaires idéales. Le principal défi, dans ce cadre là, serait de concevoir des structures qui répondent correctement à la fois pour une grande plage d'angles d'incidence et pour quasiment tout le spectre visible.

Enfin, on pourrait étudier d'autres tailles de pixel plénoptique (3x3, 4x4,...) en utilisant des sous-pixels "à l'état de l'art" c'est-à-dire d'une taille d'environ 0.6 μm . Cela permettrait de se rapprocher des possibilités d'une vraie caméra *light-field* de type Lytro ou Raytrix, tout en gardant une bonne résolution spatiale. Par exemple on forme des microlentilles de 2.4 μm pour une taille plénoptique de 4x4 en utilisant des pixels de 0.6 μm , soit un capteur 24x36 "full-frame" de 150 MPix. Ce type de capteur pourrait être utilisé pour des applications scientifiques, par exemple dans le cadre d'un télescope. L'optique adaptative permet de corriger les turbulences de l'atmosphère[67] mais il existe des aberrations optiques résiduelles. On pourrait envisager de réduire ces erreurs à l'aide d'un capteur plénoptique, utilisé directement comme imageur, et de la technique de correction des aberrations décrite dans le chapitre 4.

Bibliographie

- [1] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *European Conference on Computer Vision (ECCV)*, pages –. Springer, 2020.
- [2] Abdullah Abuolaim, Radu Timofte, Michael S. Brown, Dafeng Zhang, Xiaobing Wang, Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ling Shao, Shuai Liu, Lei Lei, Chaoyu Feng, Zhiwei Xiong, Zeyu Xiao, Ruikang Xu, Yunan Zhu, Dong Liu, Tu Vo, Si Miao, Nisarg A. Shah, Pengwei Liang, Zhiwei Zhong, Xingyu Hu, Yiqun Chen, Chenghua Li, Xiaoying Bai, Chi Zhang, Yiheng Yao, Ruipeng Gang, Sabari Nathan, Thangavelu Ragavendran, Venkatakrishnan Srinija, and Venkatakrishnan Srivatsav. Ntire 2021 challenge for defocus deblurring using dual-pixel images : Methods and results. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 578–587, 2021.
- [3] Adelson. The Plenoptic Function and the Elements of Early Vision. In *Computational Models of Visual Processing*. The MIT Press, 10 1991.
- [4] E.H. Adelson and J.Y.A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2) :99–106, 1992.
- [5] Tournier Arnaud, F Leverd, L Favennec, C Perrot, L Pinzelli, M Gatefait, N Cherault, Damien Jeanjean, J.-P Carrere, F Hirigoyen, Lindsay Grant, and F Roy. Pixel-to-pixel isolation by deep trench technology : Application to cmos image sensor. In *IISW*, 2011.
- [6] Yasir Aslam and Santhi N. A review of deep learning approaches for image analysis. In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 709–714, 2019.
- [7] M. Bigas, E. Cabruja, J. Forest, and J. Salvi. Review of cmos image sensors. *Microelectronics Journal*, 37(5) :433–451, 2006.
- [8] SCHUBERT BOREL, VANDAME. Method for correcting aberration affecting light-field data, 2017.
- [9] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4) :325–376, December 1992.
- [10] Marie-Anne Burcklen. *Conception conjointe de l’optique et du traitement pour l’optimisation des systèmes d’imagerie*. Theses, Université Paris Saclay (COMUE), February 2018.
- [11] Marie-Anne Burcklen, Frédéric Diaz, François Leprêtre, Mane-Si Laure Lee, Anne Delboulbé, Brigitte Loiseaux, Philippe Millet, François Duhem, Fabrice Lemonnier, Hervé Sauer, and François Goudail. Recent advances in joint optical-digital design for optronics applications. In Jay N. Vizgaitis, Bjørn F. Andresen, Peter L. Marasco,

- Jasbinder S. Sanghera, and Miguel P. Snyder, editors, *Advanced Optics for Defense Applications : UV through LWIR*, volume 9822, pages 92 – 101. International Society for Optics and Photonics, SPIE, 2016.
- [12] Marie-Anne Burcklen, Hervé Sauer, Frédéric Diaz, and François Goudail. Joint digital-optical design of complex lenses using a surrogate image quality criterion adapted to commercial optical design software. *Appl. Opt.*, 57(30) :9005–9015, Oct 2018.
- [13] C.-Y. Wee and R. Paramesran. Image sharpness measure using eigenvalues. In *2008 9th International Conference on Signal Processing*, pages 840–843, Oct 2008.
- [14] Canon. <https://www.canon.fr/lenses/canon-rf-800mm-f11-is-stm/>.
- [15] Marcela Carvalho, Bertrand Le Saux, Pauline Trouvé-Peloux, Andrés Almansa, and Frédéric Champagnat. On regression losses for deep depth estimation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2915–2919, 2018.
- [16] Guillaume Chataignier, Benoit Vandame, and Jérôme Vaillant. Joint electromagnetic and ray-tracing simulations for quad-pixel sensor and computational imaging. *Opt. Express*, 27(21) :30486–30501, Oct 2019.
- [17] Guillaume Chataignier, Benoit Vandame, and Jerome Vaillant. Deep learning applied to quad-pixel plenoptic images. In Daniel G. Smith, Frank Wyrowski, and Andreas Erdmann, editors, *Computational Optics 2021*, volume 11875, pages 1 – 11. International Society for Optics and Photonics, SPIE, 2021.
- [18] Pierre Chevalier. *Etude d’une méthode de micro-fabrication 3D pour des applications microlentilles d’imageurs*. Theses, Université Grenoble Alpes [2020-....], March 2021.
- [19] 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 pdaif cmos image sensor with 0.64umx1.28um photodiode separated by self-aligned in-pixel deep trench isolation for high af performance. In *2017 Symposium on VLSI Technology*, pages T104–T105, June 2017.
- [20] K. De and V. Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64 :149 – 158, 2013. International Conference on Design and Manufacturing (IConDM2013).
- [21] V. Dragoi, A. Filbert, S. Zhu, and G. Mittendorfer. Cmos wafer bonding for back-side illuminated image sensors fabrication. In *2010 11th International Conference on Electronic Packaging Technology High Density Packaging*, pages 27–30, 2010.
- [22] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. An introductory review of deep learning for prediction models with big data. *Frontiers in Artificial Intelligence*, 3 :4, 2020.
- [23] EMVA. Standard for measurement and presentation of specifications for machine vision sensors and cameras. <https://www.emva.org/standards-technology/emva-1288/>.
- [24] Eric R. Fossum. The invention of cmos image sensors : A camera in every pocket. In *2020 Pan Pacific Microelectronics Symposium (Pan Pacific)*, pages 1–6, 2020.
- [25] Eric R. Fossum and Donald B. Hondongwa. A review of the pinned photodiode for ccd and cmos image sensors. *IEEE Journal of the Electron Devices Society*, 2(3) :33–43, 2014.

- [26] Koichi Fukuda. A compressed $n \times n$ multi-pixel imaging and cross phase-detection af with $n \times 1$ rgrb + $1 \times n$ gb hetero multi-pixel image sensors. In *IJWS*, 2021.
- [27] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan Barron. Learning single camera depth estimation using dual-pixels. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7627–7636, 2019.
- [28] Todor Georgiev. Plenoptic camera resolution. In *Imaging and Applied Optics 2015*, page JTh4A.2. Optical Society of America, 2015.
- [29] Todor Georgiev, Georgi Chunev, and Andrew Lumsdaine. Superresolution with the focused plenoptic camera. In Charles A. Bouman, Ilya Pollak, and Patrick J. Wolfe, editors, *Computational Imaging IX*, volume 7873, pages 232 – 244. International Society for Optics and Photonics, SPIE, 2011.
- [30] Todor Georgiev and Andrew Lumsdaine. Multimode plenoptic imaging. In Nitin Sampat, Radka Tezaur, and Dietmar Wüller, editors, *Digital Photography XI*, volume 9404, pages 1 – 17. International Society for Optics and Photonics, SPIE, 2015.
- [31] Todor Georgiev, Zhan Yu, Andrew Lumsdaine, and Sergio Goma. Lytro camera technology : theory, algorithms, performance analysis. In Dietmar Wüller, Kevin J. Matherson, Reiner Creutzburg, David Akopian, Todor G. Georgiev, Andrew Lumsdaine, Cees G. M. Snoek, and Lyndon S. Kennedy, editors, *Multimedia Content and Mobile Devices*, volume 8667, pages 458 – 467. International Society for Optics and Photonics, SPIE, 2013.
- [32] Todor G. Georgiev and Andrew Lumsdaine. Focused plenoptic camera and rendering. *Journal of Electronic Imaging*, 19(2) :1 – 11, 2010.
- [33] A. Gershun. The light field. *Journal of Mathematics and Physics*, 18(1-4) :51–151, 1939.
- [34] Pascal Getreuer. Linear Methods for Image Interpolation. *Image Processing On Line*, 1 :238–259, 2011. https://doi.org/10.5201/ipol.2011.g_lmii.
- [35] Bhawna Goyal, Ayush Dogra, Sunil Agrawal, B.S. Sohi, and Apoorav Sharma. Image denoising review : From classical to state-of-the-art approaches. *Information Fusion*, 55 :220–244, 2020.
- [36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [37] Flavien Hirigoyen, Axel Crocherie, Jérôme M. Vaillant, and Yvon Cazaux. FDTD-based optical simulations methodology for CMOS image sensor pixels architecture and process optimization. In Morley M. Blouke and Erik Bodegom, editors, *Sensors, Cameras, and Systems for Industrial/Scientific Applications IX*, volume 6816, pages 65 – 76. International Society for Optics and Photonics, SPIE, 2008.
- [38] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [39] Matthieu Hog, Neus Sabater, and Christine Guillemot. Long short term memory networks for light field view synthesis. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 724–728, 2019.

- [40] Yu-Wen Huang, Ching-Yeh Chen, Chen-Han Tsai, Chun-Fu Shen, and Liang-Gee Chen. Survey on block matching motion estimation algorithms and architectures with new results. *VLSI Signal Processing*, 42 :297–320, 03 2006.
- [41] Diederik P. Kingma and Jimmy Ba. Adam : A method for stochastic optimization, 2017.
- [42] Masahiro Kobayashi, Masanobu Ohmura, Hidekazu Takahashi, Takahiro Shirai, Katsuhito Sakurai, Takeshi Ichikawa, Hiroshi Yuzurihara, and Shunsuke Inoue. High-definition and high-sensitivity cmos image sensor with all-pixel image plane phase-difference detection autofocus. *Japanese Journal of Applied Physics*, 57(10) :1002B5, 2018.
- [43] Edmund Y. Lam. Computational photography with plenoptic camera and light field capture : tutorial. *J. Opt. Soc. Am. A*, 32(11) :2021–2032, Nov 2015.
- [44] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.
- [45] M. Levoy. Light fields and computational imaging. *Computer*, 39(8) :46–55, 2006.
- [46] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 31–42, New York, NY, USA, 1996. Association for Computing Machinery.
- [47] Marc Levoy, Ren Ng, Andrew Adams, Matthew Footer, and Mark Horowitz. Light field microscopy. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, page 924–934, New York, NY, USA, 2006. Association for Computing Machinery.
- [48] LibRaw. <https://www.fastrawviewer.com/DPRSPLIT>.
- [49] Lippmann, G. épreuves réversibles donnant la sensation du relief. *J. Phys. Theor. Appl.*, 7(1) :821–825, 1908.
- [50] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.
- [51] O. Losson, L. Macaire, and Y. Yang. Chapter 5 - comparison of color demosaicing methods. In Peter W. Hawkes, editor, *Advances in Imaging and electron Physics*, volume 162 of *Advances in Imaging and Electron Physics*, pages 173–265. Elsevier, 2010.
- [52] Andrew Lumsdaine and Todor Georgiev. The focused plenoptic camera. In *2009 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2009.
- [53] J. Navarro and N. Sabater. Learning occlusion-aware view synthesis for light fields. *Pattern Analysis and Applications*, 24(3) :1319–1334, Feb 2021.
- [54] Ren Ng. *Digital Light Field Photography*. Phd thesis, Stanford University, CA, USA, July 2006.
- [55] Ren Ng and Pat Hanrahan. Digital correction of lens aberrations in light field photography. In G. Groot Gregory, Joseph M. Howard, and R. John Koshel, editors, *International Optical Design Conference 2006*, volume 6342, pages 441 – 454. International Society for Optics and Photonics, SPIE, 2006.
- [56] Ren Ng and Patrick M. Hanrahan. Digital correction of lens aberrations in light field photography. In *International Optical Design*, page WB2. Optical Society of America, 2006.

- [57] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light Field Photography with a Hand-held Plenoptic Camera. Research Report CSTR 2005-02, Stanford university, April 2005.
- [58] T. Okawa, S. Ooki, H. Yamajo, M. Kawada, M. Tachi, K. Goi, T. Yamasaki, H. Iwashita, M. Nakamizo, T. Ogasahara, Y. Kitano, and K. Tatani. A 1/2inch 48m all pda f cmos image sensor using 0.8 μ m quad bayer coding 2 \times 2ocl with 1.0lux minimum af illuminance level. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 16.3.1–16.3.4, 2019.
- [59] Donghyuk Park, Seung-Wook Lee, Jinhwa Han, Dongyoung Jang, Heesang Kwon, Seungwon Cha, Mihye Kim, Haewon Lee, Sungho Suh, Woong Joo, Yunki Lee, Seung-joo Nah, Heegeun Jeong, Bumsuk Kim, Sangil Jung, Jesuk Lee, Yitae Kim, Chang-Rok Moon, and Yongin Park. A 0.8 μ m smart dual conversion gain pixel for 64 megapixels cmos image sensor with 12k e- full-well capacitance and low dark noise. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 16.2.1–16.2.4, 2019.
- [60] Matt Pharr, Wenzel Jakob, and Greg Humphreys. Pbrt v2. <https://www.pbrt.org/>.
- [61] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering : From Theory to Implementation*. Morgan Kaufmann Publishers Inc., III edition, 2016.
- [62] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael S. Brown. Modeling defocus-disparity in dual-pixel sensors. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2020.
- [63] Abhijith Punnappurath and Michael S. Brown. Reflection removal using a dual-pixel sensor. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1556–1565, 2019.
- [64] Junghyung Pyo, Kyungho Lee, Eun Sub Shim, Wooseok Choi, Jungbin Yun, Taesub Jung, Kyungduck Lee, Seyoung Kim, Chanhee Lee, Seungki Baek, Junseok Yang, Jongwon Choi, Bumsuk Kim, Chang-Rok Moon, JungChak Ahn, and Duckhyun Chang. A smart dual pixel technology for accurate and all-directional auto focus in cmos image sensors. In *IWS*, 2021.
- [65] Yong Ren and Xuemin Cheng. Review of convolutional neural network optimization and training in image processing. In Jiubin Tan and Jie Lin, editors, *Tenth International Symposium on Precision Engineering Measurements and Instrumentation*, volume 11053, pages 788 – 797. International Society for Optics and Photonics, SPIE, 2019.
- [66] V Rochus, X Rottenberg, Ingrid De Wolf, P Soussan, and P De Moor. Pixel performance enhancement by integrated diffractive optics. In *IWS*, pages 1–4, 2015.
- [67] Roddier. *Adaptive Optics in Astronomy*. Cambridge University Press, 1999.
- [68] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.
- [69] Mozhdeh Seifi, Neus Sabater, Valter Drazic, and Patrick Perez. Disparity-guided demosaicking of light field images. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5482–5486, 2014.

- [70] Daniel G. Smith. *Fresnel Zone Plate*, pages 49–50. Society of Photo-Optical Instrumentation Engineers, 2013. <https://www.spiedigitallibrary.org/ebooks/FG/Field-Guide-to-Physical-Optics/Fresnel-Zone-Plate/Fresnel-Zone-Plate/10.1117/3.883971.ch48>.
- [71] George E. Smith. The invention and early history of the ccd. *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 607(1) :1–6, 2009. Radiation Imaging Detectors 2008.
- [72] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv : Neural reflectance and visibility fields for re-lighting and view synthesis, 2020.
- [73] STMicroelectronics. Img140. https://www.st.com/content/st_com/ko/about/innovation---technology/imaging.html.
- [74] Nai-Sheng Syu, Yu-Sheng Chen, and Yung-Yu Chuang. Learning deep convolutional networks for demosaicing. *CoRR*, abs/1802.03769, 2018.
- [75] Tomasz Szandała. *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*, pages 203–224. Springer Singapore, Singapore, 2021.
- [76] Pauline Trouvé. *Conception conjointe optique/traitement pour un imageur compact à capacité 3D. (Digital-optical codesign of an imaging system with 3D ability)*. PhD thesis, École centrale de Nantes, France, 2012.
- [77] Rein van den Boomgaard and Rik van der Weij. Gaussian convolutions. numerical approximations based on interpolation. In Michael Kerckhove, editor, *Scale-Space and Morphology in Computer Vision, Third International Conference, Scale-Space 2001, Vancouver, Canada, July 7-8, 2001, Proceedings*, volume 2106 of *Lecture Notes in Computer Science*, pages 205–214. Springer, 2001.
- [78] Benoît Vandame, Valter Drazic, Matthieu Hog, and Neus Sabater. Plenoptic sensor : Application to extend field-of-view. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2205–2209, 2018.
- [79] Benoît Vandame. Fast and efficient resampling for multi-frame super-resolution. In *21st European Signal Processing Conference (EUSIPCO 2013)*, pages 1–5, 2013.
- [80] Divakar Verma, Manish Kumar, and Srinivas Eregala. Deep demosaicing using resnet-bottleneck architecture. In Neeta Nain, Santosh Kumar Vipparthi, and Balasubramanian Raman, editors, *Computer Vision and Image Processing*, pages 170–179, Singapore, 2020. Springer Singapore.
- [81] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Trans. Graph.*, 37(4), July 2018.
- [82] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3) :765–776, July 2005.
- [83] Rudy Winston. Canon u.s.a., inc., whats new eos 5d mark iv dual pixel raw images. <https://www.usa.canon.com/>, 2016.

- [84] K. Yonemoto and H. Sumi. A cmos image sensor with a simple fixed-pattern-noise-reduction technology and a hole accumulation diode. *IEEE Journal of Solid-State Circuits*, 35(12) :2038–2043, 2000.
- [85] Nanfang Yu and Federico Capasso. Flat optics with designer metasurfaces. *Nature Materials*, 13(2) :139–150, Feb 2014.
- [86] Zhan Yu, Jingyi Yu, Andrew Lumsdaine, and Todor Georgiev. An analysis of color demosaicing in plenoptic cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 901–908, 2012.
- [87] Jung-Hyun Yun, Kyungho Lee, J. Pyo, Kyungduk Lee, Seungjoon Lee, M. Fujita, K. Mok, Son, Junseok Yang, Younguk Song, Hyejung Kim, Younghwan Park, Sungsoo Choi, E. Shim, Jeongjin, Cho, Seungjin Lee, Seogky Yoon, Sangil Jung, T. Nagano, Changrok Moon, and Yong-In Park. A small-size dual pixel cmos image sensor with vertically broad photodiode of $0.61\mu\text{m}$ pitch. In *International Image Sensor Workshop*, 2019.
- [88] Yinda Zhang, Neal Wadhwa, Sergio Orts-Escolano, Christian Häne, Sean Fanello, and Rahul Garg. Du2net : Learning depth estimation from dual-cameras and dual-pixels. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 582–598, Cham, 2020. Springer International Publishing.
- [89] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1) :47–57, 2017.

Table des figures

1.1	Chaîne d'imagerie conventionnelle : la scène renvoie des rayons de la source lumineuse vers l'objectif. Ce dernier forme une image sur le capteur qui l'échantillonne et l'enregistre. Les données enregistrées sont traitées informatiquement pour créer l'image finale.	1
1.2	Représentation d'un rayon avec cinq coordonnées à gauche, puis avec une paramétrisation à deux plans à droite. Illustration tirée de Lam 2015[43]. .	3
1.3	Capture du champ de lumière par la chaîne d'acquisition classique : l'information de direction est perdue. Les rayons sont intégrés de la même manière quel que soit leur angle d'incidence.	3
1.4	Exemples de diagrammes rayon/espace. La figure a) représente le cas où les rayons convergent en un même point sur l'axe x quelles que soient leurs coordonnées d'origine sur l'axe u . Les cas b) et c) représentent les cas où les rayons convergent devant et derrière le plan (x, y) considéré. Illustration tirée de Lam 2015[43].	4
1.5	Une Lytro illum.	5
1.6	Capteur plénoptique de type 1 (dit défocalisé). Ici les rayons sont intégrés par des pixels différents en fonction de leur angle d'incidence.	5
1.7	Un Canon 70D et une vue de ses sous-pixels rectangulaires. La partie en niveaux de gris est une photo prise avec un microscope électronique à balayage où l'on voit deux pixels et les transistors formant leur circuit de lecture. La vue de droite est issue d'un microscope à force atomique capacitif où l'on voit deux pixels entiers, formés par deux sous-photodiodes rectangulaires (rectangles en doré notés "PD"). Les microlentilles ne sont pas visibles, elles sont placées au dessus des groupes de photodiodes. . . .	6
1.8	Carte de profondeur générée par Google à l'aide d'un capteur dual-pixel et d'un système dual-caméras. La colonne la plus à droite est la fusion des deux systèmes.	6
1.9	Une camera Raytrix.	7
1.10	Capteur plénoptique de type 2 (dit focalisé).	7
1.11	Dans l'ordre : une caméra Pelican et la matrice de caméras d'Interdigital. .	8
1.12	Définition des images de sous-ouverture (Sub Aperture Images).	8
1.13	Illustration de la disparité.	9
1.14	Illustration du principe de la refocalisation.	10

1.15	Exemple de refocalisation d'une image. L'image a) est une mire de texte placée au-delà du plan de netteté, sans refocalisation, donc floue. L'image b) est la version refocalisée avec un paramètre $\rho = +2$, qui est maintenant nette. Les images c) et d) illustrent le cas où la mire est placée devant le plan de netteté et où l'on a utilisé un paramètre $\rho = -2$	10
2.1	Design général d'un quad-pixel	14
2.2	Réponses angulaires lors du processus d'optimisation	16
2.3	Réponses angulaires des pixels de $3.5\ \mu\text{m}$ et de $2.8\ \mu\text{m}$	17
2.4	Réponses angulaires d'un quad-pixels de $3.5\ \mu\text{m}$ et de $1.6\ \mu\text{m}$	18
2.5	Ratio 2D pour le quad-pixel de $3.5\ \mu\text{m}$ et celui de $1.6\ \mu\text{m}$	19
2.6	Chief Ray Angle	19
2.7	Chief Ray Angle Correction	20
2.8	Exemple de CRA selon la position sur le capteur	20
2.9	Exemple de barycentre de la tache de diffraction mesurée à la surface du silicium. La surface représentée en bleu fait la taille du quad-pixel soit $3.5\ \mu\text{m}$ de côté.	21
2.10	Méthode de calcul des valeurs de décalage pour le CRA	22
2.11	Réponses angulaires en 4D du sous-pixel A pour les différentes valeurs de CRA.	24
2.12	Réponses angulaires 1D pour différentes longueurs d'onde.	25
2.13	Réponses angulaires d'un dual-pixel Canon de $6.4\ \mu\text{m}$ en a) et $1.8\ \mu\text{m}$ en b). Illustration tirée de Kobayashi[42].	26
2.14	Réponses angulaires d'un dual-pixel Samsung de $0.61\ \mu\text{m}$	26
2.15	Impact de la taille de la microlentille sur les plages angulaires	27
2.16	Illustration des parties utiles et non utiles de la pupille de l'objectif	28
2.17	Schéma et réponses angulaires du pixel comprenant des barrières d'isolation	29
2.18	Résultat des simulations en sous-pixels $1.75\ \mu\text{m}$	30
2.19	Résultat des simulations en sous-pixels $0.8\ \mu\text{m}$	30
2.20	Schémas de designs alternatifs de quad-pixels $1.6\ \mu\text{m}$	31
2.21	Ratio gauche/droite du pixel avec des isolations dans la résine colorée, fabriquées avec un matériau de bas indice de réfraction.	31
3.1	Exemple de kernel produit par le modèle de Punnappurath. Illustration tirée de leur article[62]	35
3.2	Fonctions de flou simulées	37
3.3	Problème de l'optique géométrique pour la simulation de la diffraction	38
3.4	a) et b) montrent la procédure de lancé de rayon classique et la réponse angulaire idéale produite. c) et d) montrent notre méthode et la réponse angulaire réelle.	39
3.5	Description des expériences de validation de la méthode de ray tracing modifiée	40
3.6	Comparaison des réponses angulaires des expériences de validation	41
3.7	(a) <i>ML-1</i> , (b) <i>ML-2</i> , (c) ABCD texture.	42
3.8	Exemples d'images de synthèse	42
3.9	Impact de la taille des microlentilles sur les SAIs	43

4.1	Isolation d'une personne et de l'arrière-plan en utilisant l'information de profondeur d'un dual-pixels d'un smartphone Google.	45
4.2	Exemples d'images de notre base de données. a) et c) sont issues de la scène San Miguel, b) est la carte de profondeur de a), et d) est une photo personnelle placée comme texture plane.	46
4.3	Illustration du block-matching	47
4.4	Interpolation effectuée en forward (a) et en backward (b)	49
4.5	Exemple de kernels d'interpolation, dans l'ordre : plus proche voisin, linéaire, cubique et sinus cardinal. Illustration tirée de Boomgaard[77].	50
4.6	Comparaison de l'application de l'algorithme de block matching	50
4.7	Aberrations de l'optique principale vues par un capteur quad-pixel. (a) montre une aberration sur un coin du capteur où les rayons orange et jaunes ne convergent pas au même point, contrairement aux rayons bleus au centre. Les figures b) et c) représentent les microlentilles et les SAIs du quart supérieur droit du capteur. b) montre l'impact des rayons sur les microlentilles. c) montre les quatre SAIs et l'impact des rayons sur les sous-pixels. L'écart de position Δ sur les microlentilles se traduit par des disparités ρ sur les SAIs.	52
4.8	Mire de bruit blanc à gauche, bruit bleu à droite. La <i>couleur</i> du bruit fait référence à ses fréquences spatiales. Les fréquences spatiales d'un bruit blanc ont toutes la même énergie, tandis que pour un bruit bleu les hautes fréquences sont plus présentes que les basses fréquences. Illustrations tirées de http://momentsingraphics.de/BlueNoise.html	53
4.9	Résultats de la correction des aberrations pour une zone au centre et dans un coin ($\theta_{cra} = 10^\circ$ / $\varphi_{cra} = 45^\circ$). Trois cas sont présentés : objectif 1 sans diffraction (ligne 1), objectif 1 avec diffraction (ligne 2) et objectif 2 avec diffraction (ligne 3).	55
4.10	Les réseaux proposés par Syu et. al, en version simple à gauche et profonde à droite. Illustration tirée de Syu[74].	56
4.11	Le réseau proposé par Verma et. al, formé par 20 blocs résiduels. Illustration tirée de Verma[80].	57
4.12	Architecture de notre réseau de dématricage.	57
4.13	Exemple d'inférence en dématricage couleur	59
4.14	Inférence de la Figure 4.13 zoomée sur une zone au centre de l'image de sous-ouverture A. A gauche : l'image d'entrée mosaïquée, au centre : la vérité terrain, à droite : l'inférence du réseau.	59
4.15	Réseau résiduel utilisé pour la correction de la diffraction.	60
4.16	Réseau en "U" pour la correction de la diffraction. Les nombres de canaux sont indiqués dans chaque bloc. Les flèches en bleu sont des convolutions qui divisent la résolution spatiale par deux (noyau de taille 2, stride de 2). Les flèches en orange sont des convolutions transposées qui augmentent la résolution spatiale par deux. Les flèches en vert sont des connexions directes + concaténations.	60
4.17	Exemple d'inférence sur une image "brute". L'image rendue avec diffraction est à gauche, c'est celle qui est passée au réseau. L'image rendue sans diffraction est au centre, c'est la vérité terrain. L'inférence du réseau est à droite. Le PSNR entre la vérité terrain et l'inférence est de 22dB.	61

4.18	L'image brute de la figure Figure 4.17 a été dématricée avec une interpolation bilinéaire et présentée sous forme de SAIs. Elles sont placées dans le même ordre : image avec diffraction à gauche, vérité terrain au centre et inférence à droite.	61
4.19	Zoom sur la figure précédente. La colonne de gauche représente les deux SAIs gauche et droite pour l'image avec diffraction. La colonne de droite représente les deux SAIs issues de l'inférence. Les lignes rouges sont tracées en prenant la partie gauche du pied du verre comme référence. L'écart entre ces lignes représente la disparité horizontale. On remarque qu'elle est plus marquée avec la diffraction corrigée.	62
4.20	Illustration de la procédure pour générer des images sans diffraction.	62
4.21	Réseau Du2Net. La partie dual-pixel a été modifiée pour prendre quatre SAIs au lieu de deux, la partie concernant la double caméra a été laissée de côté. Illustration tirée de Zhang[88].	63
4.22	Exemple de carte de disparités calculées avec un algorithme de block matching. A gauche : la moyenne des disparités gauche et droite. Au centre : les disparités horizontales. A droite : les disparités verticales. L'unité de mesure est en pixels.	64
5.1	Les deux images de sous-ouverture d'une photo de la source collimatée.	68
5.2	Réponse angulaire d'un Canon 5D mk IV. Les figures a) et b) présentent respectivement la puissance reçue par les deux sous-pixels, normalisée à 1, et le ratio associé.	68
5.3	Schéma du prototype et description des zones de test	70
5.4	Étapes de fabrication des microlentilles.	71
5.5	Coupe transverse du prototype vue par un microscope électronique à balayage.	72
5.6	Banc de mesure en angle	73
5.7	Courbe d'optimisation de la puissance de la LED	73
5.8	Schéma des sources d'incertitudes en angle	75
5.9	La figure a) montre les réponses de gauche et de droite d'un quad simulé puis son ratio associé en-dessous. La figure b) montre les courbes associées à la mesure sur le prototype.	77
5.10	Rappels des réponses angulaires des quatre sous-pixels en simulation.	78
5.11	Réponses angulaires des quatre bandes où les microlentilles sont centrées.	78
5.12	Réponses angulaires des zones où $\varphi_{cra} = 0^\circ$ et $\theta_{cra} = 0$ à 30°	79
5.13	Réponses angulaires des zones où $\varphi_{cra} = 11.25^\circ$ et $\theta_{cra} = 0$ à 30°	79
5.14	Réponses angulaires des zones où $\varphi_{cra} = 22.5^\circ$ et $\theta_{cra} = 0$ à 30°	80
5.15	Réponses angulaires des zones où $\varphi_{cra} = 33.75^\circ$ et $\theta_{cra} = 0$ à 30°	80
5.16	Réponses angulaires des zones où $\varphi_{cra} = 45^\circ$ et $\theta_{cra} = 0$ à 30°	81
5.17	Ratio par microlentille mesuré et attendu.	82
5.18	Une partie de la mire de bruit utilisée.	83
5.19	SAIs d'une mire de bruit au point et 20mm derrière le plan de netteté. Signaux gauche/droite et haut/bas extraits d'une ligne et d'une colonne.	83
5.20	Objectif photo Mitakon 25mm F#0.95 avec un masque devant la lentille frontale. Le masque a pour but d'améliorer le contraste angulaire.	83

5.21	Scène canette sans le masque couvrant l’objectif. Une coupe horizontale est représentée à droite, sans montrer de disparités claires.	84
5.22	Scène canette avec le masque. La coupe horizontale ne présente pas de disparités claires ici non plus, malgré le masque qui devrait augmenter le contraste angulaire.	84

Liste des tableaux

2.1	Description des designs des quad-pixels $3.5\text{ }\mu\text{m}$ et $0.8\text{ }\mu\text{m}$	18
2.2	Description des décalages CRA	23
4.1	Simulation de défauts de fabrication pour l'objectif 2.	53
4.2	Gains en netteté selon plusieurs mesures.	54
4.3	PSNR of inferences for $\rho = 0$ and $\rho = 1$	59
5.1	Résumé des décalages des microlentilles en fonction du CRA.	70
5.2	Remise en ordre des pixels	74
5.3	Valeurs des paramètres permettant de calculer les incertitudes en angle. . .	76

