Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

**Dissertation**

Presented to the University of Dublin, Trinity College

in fulfilment of the requirements for the Degree of

**Doctor of Philosophy in Computer Science**

March 2022

---

# Exploration in Light Field processing and editing

**Pierre Matysiak**

**Supervisor: Aljosa Smolic**

# Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Pierre Matysiak

June 7, 2022

# Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Pierre Matysiak

June 7, 2022

## Abstract

Light fields have been used in computer science research for the better part of the last three decades, and the range of applications available is ever-growing. There exist several capture methods producing output images with varying characteristics in terms of resolution or baseline, but the usability of these setups is negatively correlated with the overall quality of the output. One such method is plenoptic cameras, easy to use and without a need for complex calibrations, and resulting in images with low resolution, small baseline, and a number of visual artefacts which are still partially unaddressed. Some software solutions exist to counteract these hardware issues, but they are limited, and it has had an impact on the majority of light field research applications.

In this thesis, we take a closer look at these types of light fields in three ways, to study ways to enhance their visual quality, to use them for the purpose of colour editing, and to compare them to more modern light field methods. First we analyse the images captured by a Lytro Illum camera and the visual artefacts affecting them. Based on this we propose a set of tools to extract those images from RAW camera data, and perform demultiplexing, white balance, colour correction and denoising on them. Second, we use these enhanced images and perform some colour editing using a method called soft colour segmentation. Third we study the possibilities of NeRF, a new method to generate light fields, and compare it with previous traditional light field methods for view synthesis and depth

estimation, to showcase the benefits it could bring for easier high quality light field capture.

À ma maman et à mon papa,

qui nous ont quittés trop tôt mais

continuent de briller dans ma mémoire.

# Acknowledgments

First and foremost, I would like to thank God. Without his benevolent influence, humanity would have never been able to achieve the technological level to formalise light fields, which are the theme of the present thesis. This simple gift makes life truly worth living to its fullest, and is an irreplaceable beam of hope in an otherwise dreadful world.

I also wish to extend my gratitude toward Prof. Aljosa Smolic, for supervising me during this PhD, and for allowing me the chance to be a part of his research team. His availability and ability to guide me was more than welcome, as I easily tend to be excited and distracted by too many ideas at once, which often proved quite inefficient.

I want to thank my family for supporting me during this journey, and especially for patiently listening to me explain my work to them in as simple terms as I could formulate, which often led to blank stares and polite nods, clearly a failure on my part.

My dad was not so lucky to see me bring this thesis to its completion, but his pride in my endeavour was always a very strong motivator. His undying love and support to the very end will stay with me forever, and I can only hope to be one day as good a role model as he was for me.

The last four years were made more bearable thanks to the amazing group

of PhD candidates I had the pleasure to work with. The emulation from seeing them busy as bees was detrimental in my being able to accomplish anything, and our late-night, obviously work-related, conversations remain some of the more memorable and pleasant moments of my time spent here.

And finally a heartfelt thanks to Rachael, who has tolerated and supported me for the past year, and played a critical role in my acceptance of the worth this work holds.

Pierre Matysiak

*University of Dublin, Trinity College*
*March 2022*

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

In this chapter, we aim to introduce the reader to the domain of Light Fields and the wonderful applications and opportunities they offer, as well as drawbacks and limitations. We additionally discuss the aim of this thesis, the motivation behind the presented work, the solutions we offer, and the structure of this thesis.

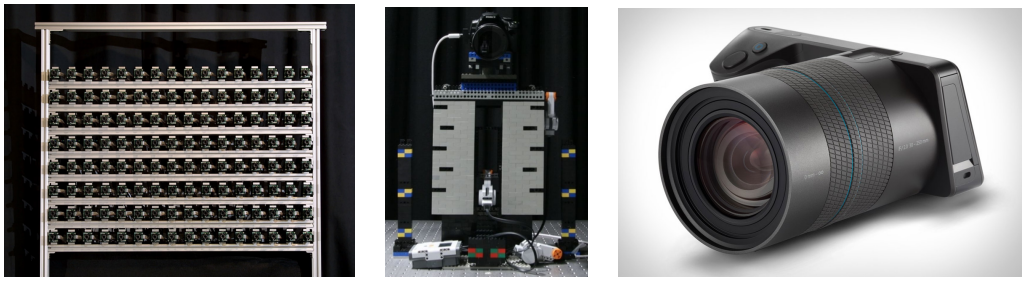


Figure 1.1: Three devices to capture light fields, the Stanford camera array (left) [11], the Stanford Lego gantry (centre) [12], and a Lytro Illum camera (right) [13].

## 1.1 Motivation

### What are light fields used for?

Light fields are, broadly put, a way of representing the light information contained in a finite volume of space [10]. They are an extension of traditional single view images and are typically represented by the 4D plenoptic function with two spatial and two angular dimensions. The most common methods of capturing real light field images are camera arrays [11], single cameras on a moving gantry [12], or consumer-grade plenoptic cameras [13] (see Figure 1.2). They allow to obtain a representation of the light field, generally in the form of sub-aperture views. All these concepts will be explained in further detail in Chapter 2 of this thesis.

A large body of research pertaining directly to light field theory and its applications exists, the field is vast and covers a wide array of tasks such as rendering, depth estimation, super-resolution, compression, novel view synthesis or coding, to name a few. However one field in particular seems to receive less attention: image editing. We posit that editing could vastly benefit from the higher dimensionality of light field data, when compared to traditional 2D methods, and this is the context in which we place the work of the present dissertation.

The reasons light field editing could be useful are three-fold: first we think it finds uses in professional movie post-production, as this has already been explored by Trottnow et al. [14]. Secondly, in generating engaging virtual reality content, and thirdly, perhaps more pragmatically, as a method for data augmentation to improve machine learning training.

### What are the current drawbacks?

Traditionally light field research has focused on the more easily captured data, which also happened to be the more widely available at the time. For the past decade most of the focus was on Lytro images and the very popular Stanford gantry dataset [12], both of which produce arrays of sub-aperture views with relatively low resolution and relatively small baseline between views. This has severely guided most research applications toward these types of images in particular, and as a result most of it is not easy to generalise to any light field image with different characteristics, such as wider baseline for instance. As we move toward more easily available high resolution high baseline light field data, it is paramount to also ensure the software for existing applications can be used for these equally well. The work presented here does for the most part, as only colour consistency and accurate depth maps are needed for the applications we propose.

We put in this thesis our gaze toward one of the most easy to capture light field data, Lytro images. As they are the result of a photograph taken from a single viewpoint, taken using a unique, albeit advanced, camera, they suffer from a number of limitations, caused by physical limits in the manufacturing of the camera itself, which affect the visual quality of their output (see Figure 1.1 for a visual example). In particular this affects negatively the sub-aperture views placed on the outer edges of the light field. Therefore, and traditionally in Lytro-based research, most teams tend to ignore those outer sub-views. This usually represents over half of the available sub-views for a particular image, thus reducing substantially the amount of information the medium was meant to provide. In the next section we present a selection of such problems we studied in this thesis.

## 1.2 Problems of Light Field Images

We particularly look at a few ways to better use light field images, and focus on quality enhancement, as well as editing applications.

Figure 1.2: Two views extracted from an image capture with a Lytro Illum and processed with the toolbox of Dansereau et al. [5]. Note the natural colours of the flower are pink and white. The colour and brightness differences between the centre view (left) and a corner view (right) are flagrant.

### Quality enhancement

As mentioned in Section 1.1 we want to look at many of the issues that affect Lytro images, as they are still among the most widely used for light field research, and as far as we are aware no research has been performed with the intent to restore the quality of sub-aperture views and therefore allow their use in research applications. The limited space in which to put the micro-lens array in the camera leads to distortion artefacts growing in importance once we reach the edges of the sub-aperture view array. Lower amounts of light hitting the camera sensor for these views further enhances these issues, or creates new ones, such as inconsistencies in brightness or colour information between the views. Additionally, the sensor quality is low, and this results in several types of noise affecting all of the sub-aperture views, which need to be corrected.

In Chapter 3 we propose a series of measures bundled into a single pipeline for processing Lytro RAW data, from the demultiplexing to the final denoising, while also looking at fixing colour inconsistencies. We thoroughly explain our process in doing so, compare our output with previously available tools for extracting Lytro data and show with a number of application the benefits that our solutions provide.

### Colour Editing

After improving the quality and usability of some of the most widely available data, we look at some of the applications that were less popular in re-

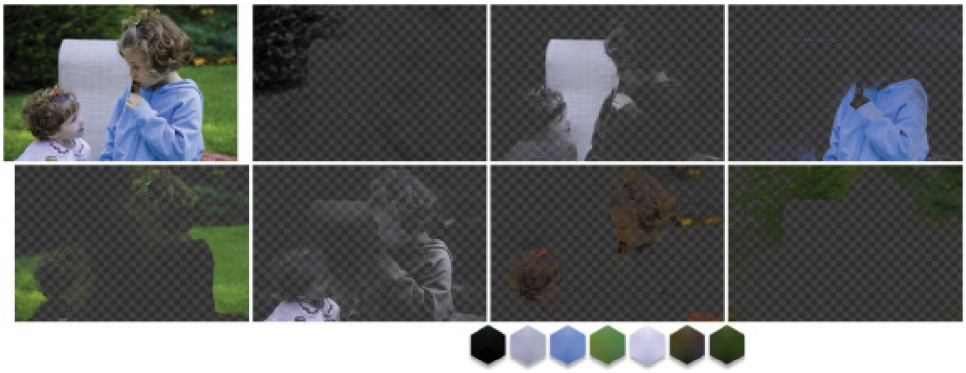


Figure 1.3: Soft colour decomposition of an image using the method of Aksoy et al [17]. Note some of the layers contain pixel information from both the girls' faces, as well as the chair. Any attempt at editing that layer would affect all these objects to a certain extent.

search. A survey of light field editing by Jarabo et al. [15] shows that while efficient, the proposed methods still require a certain degree of user interaction to perform edits. At the same time we were looking at more recent techniques for 2D image editing, such as soft-colour segmentation proposed by Aksoy et al. [16]. It extracts colour information of an image based on a computed palette and generates several semi-transparent layers containing only the pixel information of those colours. If we look at Figure 1.3 we notice that several of those layers contain information for several unrelated objects. Performing a colour edit on one of those layers will affects all objects equally, which may not be desirable.

In Chapter 4 we look such applications, and attempt to provide some initial solutions to these problems. In particular we look at the different ways it is possible to perform colour editing on part of the image or part of an object. We do so by first performing soft colour segmentation of the entire light field, which allows to retrieve information from even objects occluded in some views, and before moving on to editing, perform a depth-based object segmentation task to remove the risk of unwanted editing artefacts. We also show that both of these tasks take advantage of light field data to perform better when compared to single view editing.

### View synthesis and depth estimation

The majority of the work on light field at the time of starting this thesis was done using traditional computer vision or computer graphics tech-

niques. As time went on, machine learning became a dominant force in research work, and, while not necessarily rendering all previous work obsolete, nearly every application saw massive benefits in the use of neural network. In particular, one of them, Neural Radiance Fields (NeRF) [18] made its entrance in 2020 and generated plenty of excitement in the field, for the new possibilities it offered in terms of view synthesis. While this was a definite step in the direction of higher quality light fields, we felt the need to ensure that all the previously collected data, from any type of capture device, could still be used efficiently.

To this end, in Chapter 5, we take a closer look at NeRF and the possibilities it offers. In particular we focus on the tasks of view synthesis and depth estimation, natively performed by NeRF. The latter was especially interesting for us as our work in Chapter 4 is highly dependent on having high quality depth maps. We compare NeRF to traditional state of the art light field methods, and apply it to a selection of various light field types, to evaluate if NeRF could really become an undisputed replacement, or if the previous research could still have its place in the field.

## 1.3 Research Question and Contributions

Here we present the contributions developed in this thesis and the question that bind them all.

**Research Question**

*We attempt to investigate* — "***How can we benefit from Light Field images to perform high quality processing and editing?***".

Three main objectives are explored in this context:

- Lytro Image Quality Enhancement.
- Soft Colour Segmentation on Light Fields.
- Expanding the range of usable data using NeRF.

Contributions

- We present a pipeline for **quality enhancement** of Lytro images. We follow a thorough approach and go from demultiplexing the RAW data into usable sub-aperture views, to colour correction, hot pixel removal and denoising. We compare our results, and show definitive improvements upon the previous state of the art extraction toolbox.

- We present a software solution to apply **soft colour segmentation on light field images**, for the purpose of colour editing. We show that by using the multiple views we benefit somewhat in the quality of the palette computing and subsequent colour layer creation. We also show by using multiple view points we can extract objects based on their depth to further segment the images and perform targeted editing in an easier manner.

- We present a small **comparative study** between traditional light field methods and NeRF on the applications of novel view synthesis and depth estimation. We show that while NeRF performs well, it comes at the expense of mandatory data manipulation for it to be usable on certain type of light field data.

## 1.4 Publications

**Publications Based on Thesis Work**

- Pierre Matysiak, Mairéad Grogan, Mikaël Le Pendu, Martin Alain, Aljosa Smolic, **A Pipeline for Lenslet Light Field Quality Enhancement**, IEEE International Conference on Image Processing, October 2018 (ICIP), Athens. [19]
- Pierre Matysiak, Mairéad Grogan, Mikaël Le Pendu, Martin Alain, Emin Zerman, Aljosa Smolic **High Quality Light Field Extraction and Post-Processing for Raw Plenoptic Data** IEEE Transactions on Image Processing, 2020. [20]
- Pierre Matysiak, Mairéad Grogan, Weston Aenchbacher, Aljosa Smolic, **Soft Colour Segmentation On Light Fields**, IEEE International Conference on Image Processing, October 2020 (ICIP), Abu Dhabi. [21]

**Publications To be Submitted**

- Pierre Matysiak, Susana Ruano Sainz, Martin Alain, Aljosa Smolic, **A Comparative Study between Traditional Light Field Methods and NeRF**, IEEE International Conference on Image Processing, October 2022 (ICIP), Bordeaux.

## 1.5 Dissertation Structure

The thesis is presented as follows, and adopts a fairly classical approach. We introduced the motivation and problems in this first chapter, as well as laid down the research question and contributions proposed. The second chapter will detail what light fields are, from its historical form and evolution down to current state of the art, as well as capture methods, usage, and modern forms. The third chapter will focus on our first contribution, a method to enhance the quality of lenslet light field images, from the decoding step of the RAW images until post-processing improvements on colour and noise. The fourth chapter explores a method to perform

soft-colour segmentation on light field images, for the purpose of editing. The fifth chapter looks at a new technology to generate light fields, NeRF, and we study its appeal, advantages and drawbacks, and compare it to traditional light field methods for view synthesis and depth estimation. Finally we conclude in the last chapter with a summary of the thesis, and a discussing about what it offered, and what venues it opens for future work.

# Chapter 2

# Background

This chapter is divided in several sections in which we give an overview of light fields as a concept, and a practical application. In the first section we go over the history behind the concept of light fields - inception and evolution - and give an overview of what modern light fields are representing. Secondly we look at different methods for capturing light fields, and for each discuss the advantages, downsides, and usability. In the third section we look at use cases outside of pure research, prospects, and potential future applications. Finally we describe briefly a new method for capturing or generating light fields, which is transforming the pre-existing landscape in research, capture and applications.

## 2.1 History and description

This section aims to be a introductory guide to light fields, from its inception to its modern description and formalisation.

### Brief History of Light Fields

The first recorded studies on light and its nature date back to the early Hellenic period. The most influential theory comes from Plato who posited that light emanates from our eyes in the form of rays, and allows us to sense the shape, size and colour of every object surrounding us. While amusing, this theory still went on to be the dominating one for well over a thousand years, until it was disproven in the early $11^{th}$ century by Alhazen, who not only highlighted the idea that light rays were traveling to the eye, but also the role of the brain in interpreting these rays as coherent images [22].

It took another several hundred years, until 1846, for another level of understanding to be reached. That year Michael Faraday proposed, in a lecture on the structure of the æther named "Thoughts on Ray Vibrations", that light should be interpreted as a field occupying not only space but also time. His idea stemmed from his work on magnetism and was a theorised extension of the notion of gravity [23].

James Clerk Maxwell provided the formalization to this theory three decades later through his famous set of equations. Others have made major contributions to our understanding of the properties of light [24]. Pierre Bouguer was the first to calculate the amount of light loss when passing through the atmosphere, and discovered what we know today as the Beer-Lambert law [25]. Johann Lambert further refined the notion that light wanes with distance and time, and introduced the concept of perfect diffusion which gave its name to Lambertian objects [26]. These works and others generated widespread interest in theoretical photometry work in the first half of the $20^{th}$ century, culminating in two major achievements. The first, published in 1939 by Andrey Aleksandrovich Gershun and coining the term, was *The Light Field*, in which he presented his work on room lighting [27]. The second, in 1950, is Subrahmanyan Chandrasekhar's

seminal book, Radiative Transfer, in which he details his theory regarding polarised light [28, 29, 30].

James Kajiya introduced this work to the computer graphics literature in 1986 in his widely cited paper [31], before others, and most notably Levoy et al. took a thorough approach to describe light fields and bring them to the computer graphics field under the name *image-based rendering* [10]. This interpretation is commonly represented by using a simplification of the plenoptic function proposed by Adelson et al. [32] into a four-dimensional function describing spatial and angular dimensions. Through this function it is now possible to represent with high accuracy the light information representing a specific scene, and use that information for a variety of tasks.

### The Plenoptic Function

The plenoptic function described by Adelson et al. [32] assigns a radiance value to rays propagating within a defined physical space and is the foundation of modern light field research. It describes the light rays propagating in all directions and interacting with all objects in the 3D space, leading to occlusion, attenuation, diffraction, and all manners of alteration.

The plenoptic function does not rely on an underlying model and is rather a phenomenological description of the light passing through space. It accommodates for all the possible variations of light and adopts a high-dimensional description by assigning arbitrary radiance values at every position of space, for every possible direction of propagation, for every wavelength, and for every point in time. This is formalised in the following equation:

$$l_\lambda(x, y, z, \theta, \phi, \lambda, t) \tag{2.1}$$

In this function, $l_\lambda[W/m^2/sr/nm/s]$ describes spectral radiance per time unit, (x,y,z) is a spatial position, $(\theta, \phi)$ is an incident direction, $\lambda$ is the wavelength of light, and $t$ is a temporal instance [33].

The plenoptic function is mostly of conceptual interest. It is an idealised

function which can not be directly expressed in computer vision or graphics terms, and therefore needs to be adapted for the purpose. For instance, since radiant flux - or light - is being delivered in quantised units, *i.e.* photons, it needs to be measured during a time average, rather than an instantaneous snapshot. Similarly, we are physically constrained by the physical world and measuring infinitely thin pencils of rays is not possible without seeing artefacts and wave effects. This reduces the type of frequencies that can be measured, and restricts the scene to macroscopic settings with objects significantly larger than the wavelength of light [33].

Modern Light fields derive from the plenoptic function by introducing additional constraints [33]:

- They are considered to be static, even though video light fields have been explored [11] and are becoming increasingly feasible. An integration over the exposure period removes the temporal dimension of the plenoptic function.
- They were initially considered to be monochromatic, but the reasoning can be applied to each color channels independently. An integration over the spectral sensitivity of the camera pixels removes the spectral dimension of the plenoptic function.
- Finally, the "free-space" assumption, that the viewpoints are outside the convex hull of the scene, introduces a correlation between spatial positions. Rays are assumed to propagate through a vacuum without objects, except for those contained in an "inside" region of the space, often called a scene. Without a medium and without occluding objects, the radiance is constant along the rays in the "outside" region. This removes one additional dimension from the plenoptic function [32].

A light field can therefore be described as a four-dimensional (4D) function, interchangeably called *4D light field* by Levoy et al. [10] or *Lumigraph* by Gortler et al. [34], and is defined as the radiance along rays in empty space [10]. Typically in computer graphics this function is parameterised as $L(u, v, s, t)$ with *s* and *t* representing the spatial dimensions, and *u* and

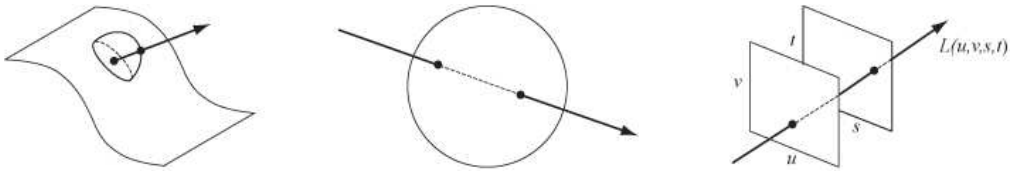



Figure 2.1: Alternative parameterisations of the 4D light field. On the left, points on a curved surface and directions leaving each point. In the middle, pairs of points on the surface of a sphere. On the right, pairs of points on two planes in general position. [30]

*v* representing the angular dimensions. Essentially, this means we can interpret a 4D light field as a collection of perspective images on the *st* plane taken from a viewpoint on the *uv* plane, see Figure 2.1. In layman's terms, this is similar to taking photographs of a scene from different viewpoints all situated at an identical distance from the scene. These collections of images can be processed into obtaining a continuous view of the underlying scene, which is the principle behind the concept of *image-based rendering*, or in this case, *light field rendering*.

### Light Field Rendering

*Image-based rendering* is a collection of techniques to represent an object or scene on a computer display using images of that scene previously captured by a camera, rather than by creating a 3D model of the object manually. The idea was first proposed by Eric Chen [35], in Apple's software QuickTime VR. As seen in Figure 2.2 it is composed of a collection of images captured all around a centre object, which allows the user to freely navigate around the object, though not toward or away from it, at a later time. However, if the views are captured densely enough, it is possible, by selecting among the pixels of a small subset of neighbouring views, and possibly using interpolation among these pixels, to generate novel, perspectively accurate views from positions where no observation was made. This is called *light field rendering* [30].

This allows to generate a potentially infinite number of views around the object, from any distance so desired, so long as one stays outside of the convex hull of the object or scene of interest. Formally we can interpret a light field as a 2D collection of 2D images, or a 4D array of pixels, as illustrated in Figures 2.2 and 2.3. Any novel view is simply a correct extraction

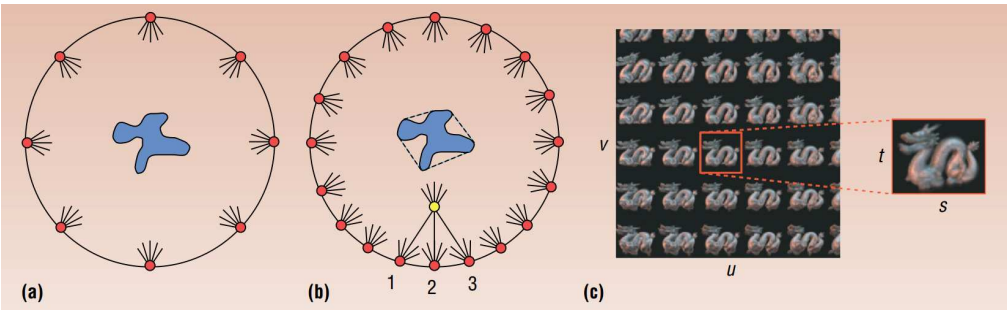



Figure 2.2: QuickTime VR setup (left). Red dots indicate camera positions. If enough real cameras exist, it is possible to interpolate their pixel information to generate synthetic views (yellow dot), this is the base of light field rendering. On the right, a light field interpreted as a 2D array of 2D images [30].

of a 2D slice from this 4D array. The amount of images necessary to render a light field depend on the application, but also the level of detail needed, as well as the locations expected for novel views. If one needs to be able to turn around all sides of an object, photographs of its back are needed. If one wants to move very close to the object and analyse its fine structure, the images necessarily need to have high spatial resolution, to be able to capture these details. The methods describing these are called the "sampling" of the light field [30].

Some amount of research has been made on light fields sampling [36, 37, 38, 39, 40, 41], and their findings agree on basic principles. If the images have low spatial resolution, the light field renderings will suffer from blur, even more so as one moves away from the original positions of capture. If the number of images is too low, the renderings will suffer from artefacts, or ghosts, arising from blending different views of an object. This is summarised using the plenoptic sampling curve, seen on Figure 2.4 [36]. On the other hand, it is possible to increase the quality of the renders by first generating a 3D model of the scene. In that case, if we take the concept to an extreme, one can reconstruct an accurate model of the scene and fly freely around it to render novel views, using potentially only a handful of images. One such method will be described in more detail in Section 2.4.

Now that the basic theoretical foundations of light fields are laid out, in the next section we will look at the various techniques used to capture them, and discuss the advantages and issues pertaining to each.

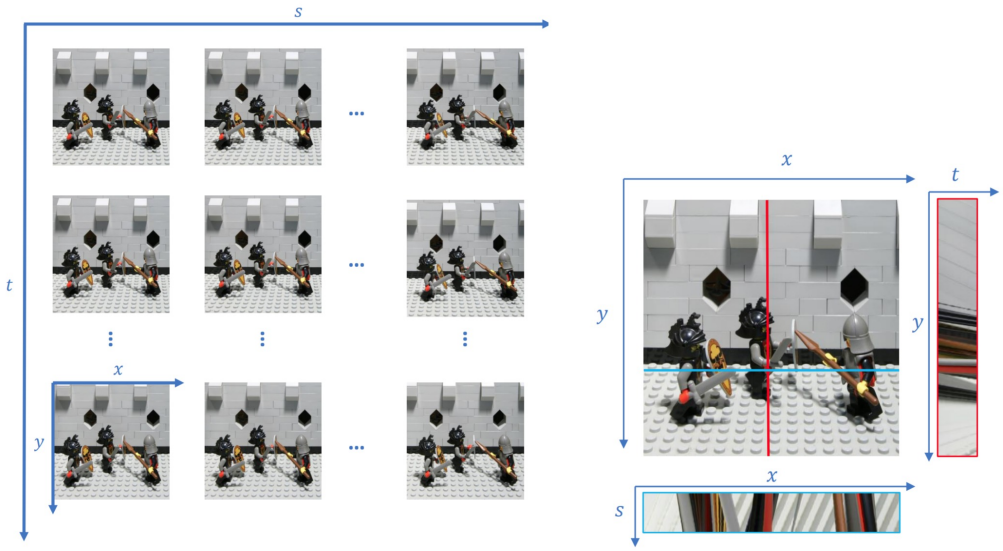



Figure 2.3: A light field can be interpreted as a 2D collection of 2D images, each captured from a different viewpoint (left). A different parameterisation of the 4D light field allows to obtain Epipolar Plane Images (right) in which the angle of the slopes are indicative of the depth of the corresponding structure.

## 2.2 Methods of capture

In this section we detail the different type of light field capture methods, namely the gantry, the camera array, the plenoptic camera, and other more recent or more niche methods.

### Camera Gantries

Perhaps the most intuitive method consists of using a single camera, and using it to capture all the images we deem necessary for obtaining a high quality light field. Done by hand, this would result in a very chaotic set of images which, to be processed effectively, would require to know the exact position each image was taken from, a logistical nightmare. To counteract these issues, the most common procedure is to strap a single camera to a rig, robotic arm or gantry, in order to have full control on its position before capturing each image.

Several examples of such structures exist. One of the first one was the Stanford Digital Michelangelo Project [42], in Figure 2.5, in which they describe a complex setup comprising not only of hardware solutions, includ-

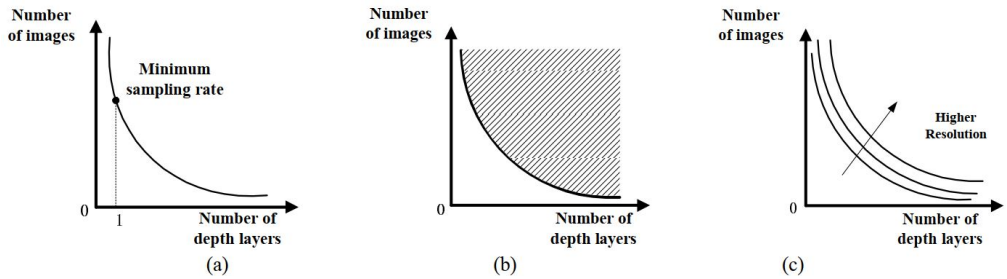



Figure 2.4: The plenoptic sampling curve. In (a) is shown the minimum sampling rate in image space. In (b) the minimum sampling curve in the joint image and geometry space (any sampling point above the curve is redundant), (c) minimum sampling curves at different rendering resolutions [36].

ing a motorised gantry assisted by a laser triangulation scanner to accurately record camera positions, but also software to be able to process the images. They tried to make capture as repeatable as possible, which allows for "easy" calibration, however physical reality meant that this process was much harder than initially anticipated. Another rig, which we briefly mentioned before, was designed by Apple and involves a camera looking inward and moving across the surface of a cylinder, which generated the QuickTime VR datasets [35], see Figure 2.2. Additional circular rigs include the Stanford Spherical Gantry, in Figure 2.5, based on the same concept [43], and the Microsoft Research/China rig which has instead a camera pointing outwards, and was developed to construct so-called "concentric mosaics" [44].

Perhaps the most popular and famous of these contraptions is the Stanford Lego Gantry [12], built using Lego Mindstorms motors, which have rotary encoders, and was made to be very accurate and repeatable, almost as precise as their first gantry, a seen in Figure 1.2. Here Lego calibration objects were included in every scene (and then cropped out) to perform calibration at the time of capture. It resulted in iconic images still widely used today for a number of light field applications.

One of the drawbacks of such systems, although shared by most methods, is the calibration. The position in space of the camera related to the scene has to be known precisely for any processing of the images to be done. This can result in tedious manipulations before and during shooting to perform accurate calibration. Additionally, these systems can only

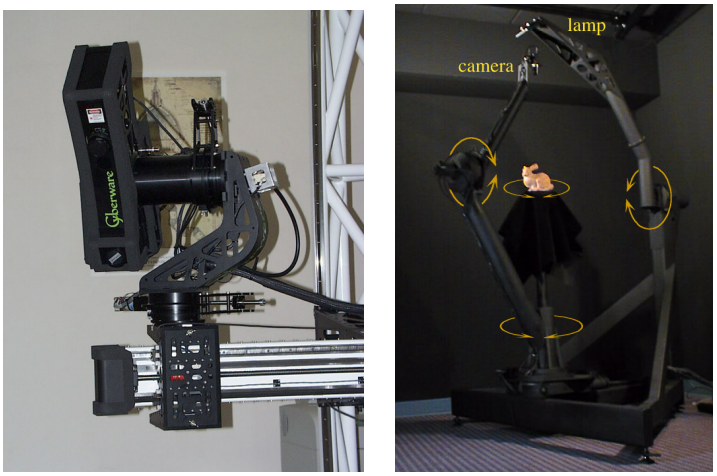



Figure 2.5: Some light field gantries. From left to right: Stanford Digital Michaelangelo camera gantry [42], Stanford Spherical gantry [43].

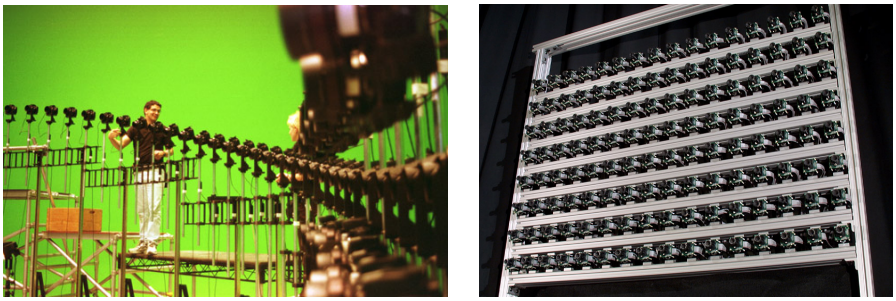


Figure 2.6: Various light field camera arrays. From left to right: Row of camera used for the slow-motion scenes in the Matrix, Stanford Camera array [11].

be used to capture static scenes, as they use a single camera which needs to be moved to each viewpoint. Capturing videos with such a structure is therefore impossible, with the exception, perhaps, of stop-motion videos, in which the motion is obtained as a result of combining a series of static images where the scenes is carefully and physically manipulated.

### Camera arrays

To alleviate some of the constraints of the gantry, it is possible to imagine putting a number of cameras in series, which would all capture the same scene at exactly the same time. The concept was introduced by Dayton Taylor in 1999 in the movie *Matrix* [30], in which cameras positioned along a 1D path were used, seen in Figure 2.6, and displaying the image they

captured in rapid succession allowed to simulate an impression of orbiting around a still scene. However, while impressive for the time, this method was just a visual trick and did not generate true light fields, as these images were never manipulated to obtain novel views [30].

However, the first attempt to generate a precise light field camera array was made by Wilburn et al. [11], seen in Figure 2.6, using a cluster of 100 inexpensive cameras mounted to a rigid array, where the placement of cameras can be reconfigured depending on the application. It was designed to capture synchronised videos in real-time, estimate 3D scene geometry, and as a novel way to construct multi-perspective panoramas. The spacing between cameras allowed them to either simulate a single-center-of-projection synthetic camera, a single camera with a large synthetic aperture, or a multiple-center-of-projection camera, which captures a light field.

Such a system alleviates one of the constraints gantry setups suffered from, which is the physical calibration of the camera positions. With such a device, this calibration is built-in and can easily be repeated for several shoots. The other limit of gantry setups is also solved here, since both photo and video capture are possible, provided all the cameras are perfectly synchronised with one another. The limit here however is the need to perfectly and equally calibrate each camera in the exact same manner. Failure to do so will result in brightness and colour inconsistencies between views, which will need to be corrected in post-processing. Using the same model camera for the entire array reduces those variations, however it is nearly impossible to negate them altogether.

Another example of camera array was developed by the SAUCE Project with the intent to use for advanced movie post-production [45, 14], using state of the art calibration and post-processing tools. They constructed an array composed of 64 high resolution cameras, allowing them to capture so-called 5D light fields, (4D rays plus time), since the exposure time of the cameras can be controlled individually. Here as well the calibration process has to be extremely precise, and is done using a long process where calibration patterns are captured from 150-200 positions, to account from between-camera occlusions, and consists on individual calibration, as well


Figure 2.7: Various plenoptic devices. From left to right: Lippmann's integral photography device, showing here the front with its twelve lenses [46], Raytrix R11 3D light field camera [47], (h) K|Lens light field lens [48].

as calibration of each possible camera pair as a stereo camera. One of the other area where particular attention was needed is the output data processing, as this array generates a very large amount of data, around 135 GBit/s. Highly efficient compression schemes were necessarily to make this output manageable while maintaining good overall image quality. While of very high quality, the output images, by nature of using different cameras, still can exhibit some colour inconsistencies between views of a single frame, and need to be corrected in post-processing.

### Plenoptic Cameras

The two previous methods described are producing high quality images, of either still or live scenes, but they are bulky, expensive, and impractical to manipulate for the average end user. Additionally, the heavy reliance on accurate calibration to obtain usable images, and the logistics necessary to process the large amount of data mean that these setups are reserved for high-end research or studio use. The only way to counter these issues would be to offer a single, hand-held device that can still somehow capture an image from multiple viewpoints.

Perhaps the first example of this was developed in 1908 by Gabriel Lippmann, in what he called "integral photography" [46]. Based on insect eyes, his device consisted of a plane array fit with tightly spaced small spherical lenses, allowing to capture a scene instantaneously from different view-

points. To view the image, one had to use a similar array of lenses, which gave the viewer the impression of a “living image”, based on the position of the eye in respect with the lens array. Theoretical at first, the concept was limited by the technology of the time, Lippmann achieving by 1911 his best system using twelve lenses, seen in Figure 2.7), but lacking the desired optical quality [49]. It nevertheless went on to become the basis of lenticular imagery [50].

Almost a century later saw the emergence of plenoptic cameras, that contain an array of micro-lenses between the main lens and the sensor, the first of which was proposed by Ng et al. [13], in Figure 1.2. Such a camera produces very dense light fields, i.e. with many views very close to one another. Plenoptic cameras have gained interest about a decade ago, after the release of two models by the Lytro company in 2006 which aimed at allowing professionals and amateurs alike to capture light fields with a dense angular sampling, where each micro-lens outputs unfocused micro-images. Unfocused cameras are generally exploited by extracting sub-aperture images, each with a very wide depth of field and representing different viewpoints of the scene. These cameras were sold on the promise of easily refocusable images. Shortly after, another similar type of camera called plenoptic 2.0 was developed by Lumsdaine et al. [47], seen in Figure 2.7. As opposed to the previous design, here each micro-lens outputs a focused micro-image. Focused cameras are typically used to render focused images where the focus can be dynamically adjusted based on user input.

Plenoptic cameras offer a number of advantages, as they are easily carried around, do not necessitate calibration for each scene, and were, at the time they were still available (the Lytro company has since closed its doors), affordable to end users and researchers. Thanks to this availability, the number of datasets available for use in research is very large. Moreover, because the images it produces are small in size, the output data is relatively convenient to process, compared to high resolution camera arrays. Finally, since the baseline between resulting views is very small, it easily lends itself to an array of research applications, such as depth estimation or view synthesis.

However due to their unique design, plenoptic cameras generate much more complex RAW data compared to traditional cameras, and the exploitation of this data is made more difficult as a result. In addition, the small design of the camera means it hits physical limits of the optics, and the result, in the case of the Lytro camera, is an output of low resolution views that also suffer from a number of inconsistencies between views, notably related to colour, as well as noise. While some of these issues were alleviated by using the provided Lytro software, it is at the time of writing not available anymore to consumers. Additionally, the Lytro software, while fixing some of visual artefacts, was unable to provide the user with sub-aperture images, which are very useful in a number of research applications. For all these reasons and more, we provide in this thesis a number of software solutions to palliate the low quality of Lytro images. These range from RAW image demultiplexing to colour correction and denoising, and are presented in great detail in chapter 3 of this thesis.

Worthy of mention in this section, the K|Lens One [48] is a very recent addition to the arsenal of tools capable of capturing light fields, and can be seen in Figure 2.7. It is the first light field lens, and can be used with any full-frame camera, thus allowing photo and video capture. The lens features an "Image Multiplier", a mirror system, which results in 9 perspective views being captured by the camera sensor. While this technology is really exciting as it provides with much higher resolution views than plenoptic cameras, albeit a lower number of them, it was not available to us at the time of writing this thesis. However we are seriously looking forward to the possibilities it could offer for casual and professional users.

### Other methods

More marginally, we can talk about many modern smartphones, containing 2, 3 or more cameras, which allow to capture light fields very similar to a camera array with a very small baseline, some examples in Figure 2.8. Most of these are used to assist the user into manipulating or improving the focus or the colour balance of their image, thus resulting in the most æsthetically pleasing output.

Figure 2.8: Other methods to capture light fields. From left to right: smartphone cameras for the iPhone 12 and Nokia 9, Zeiss/Raytrix light field microscope.

Finally, and to be thorough, we should also mention the existence of light field microscopy [51], with an example seen in Figure 2.8. In essence, it functions the same way a plenoptic camera does, as a microlens array was added to the optical train of a regular microscope, and allows to generate perspective views of biological samples or specimens. However, this method is severely constrained as diffraction puts limits on the combination of spatial and angular resolution achievable by such systems.

## 2.3 Usage of Light Fields and Prospects

In this section we discuss some practical applications for light fields, their potential use in different fields, advances and advantages they could provide. Two main avenues of applications for light fields outside of pure research are virtual reality and movie post-production, but there also exists a number of alternative research of industrial uses.

### Virtual Reality

In the last few years, virtual reality (VR) has made a resurgence, aided significantly by the improvements in computational ability of modern computers. Current generation of VR displays use optical techniques to trick the user into thinking they are looking at a distant image, video or game containing depth, when they are in fact only looking at a 2D screen, sat mere inches from the eyes. For some people, this is not necessarily an issue, as their brain can accommodate for this discrepancy and mostly believe what it is shown. However for some, this is not possible and the outcome is potentially severe side-effects: visual discomfort and fatigue,

eyestrain, diplopia (double vision), headaches, nausea, compromised image quality, as well as potentially pathologies in the developing visual system of children.

In an attempt to reduce these side-effects, VR headsets are armed with a list of visual depth cues, namely binocular disparity, motion parallax, binocular occlusions, and vergence. Binocular disparity stems from the fact both our eyes see slightly different scenes, and as our brain merges those together, the disparity between the two creates the sensation of depth. Motion parallax means that, when one moves their head, objects that are nearer move faster, or more, than objects in the distance. In VR this is used to make you think some objects are further away than they actually are, since they are still displayed on the same screen. Binocular occlusions refers to the fact that foreground objects are nearer, and therefore hide parts of background objects, which allows for a simple distance ranking. If the occlusions are different for each eye, the brain sees this as depth. Finally vergence refers to the rotation one's eyes perform to keep objects close (convergence) or far away (divergence), at the centre of one's field of vision. The amount of rotation is used by the brain, to perform simple triangulation and estimate the distance to the object.

All of these tricks can be put into effect by showing two separate images to each eye, provided they are properly synced and calibrated, and give the illusion of depth to the viewer. Despite these efforts, the brain can still sense that whatever it is shown is not the kind of reality it is used to. The reason for this is because in the brain, vergence is coupled with an additional depth cue: focus. Our eyes converge a bit to be able to focus on nearer objects, and they diverge slightly when attempting to focus on objects far away. The issue here is that VR displays do not provide any focus cues, as the entire image is always in focus, for obvious display reasons. And since our eyes still converge and diverge when they look at different parts of the scene presented to them, based on their perceived distance, it creates a perception issue named the "vergence-accommodation conflict".

In an attempt to further improve the quality of use of VR headsets, Huang et al. [52], improving on the original stereoscope designed by Sir Charles

Wheatstone in 1832 [53], created a Light Field Stereoscope, by adding a modern light field display to the classic design. This is an attempt to counteract the vergence-accommodation conflict, inherent to all stereoscopic display technology. Their model provides high resolution images, as well as additional focus cues: retinal blur, and accommodation. The light field display consists of 2 semi-transparent LCD screens over a backlight, and the combination of these recreates a light field image, which allows near-correct focus cues. Simply put, the eye naturally focuses on the part of the image it looks at, while the rest of the image looks blurred, similar to natural vision. Moreover, as the entire light field is available in the display, all of this is possible without eye tracking.

Additionally, this method increases the accuracy of monocular occlusions in complex scenes with many objects, and particularly with dark objects partially occluding bright objects. As it is not possible to block light going through the LCD displays, this results in those dark objects appearing semi-transparent, an effect not dissimilar to natural vision.

Overall, this is very promising technology and a clever way to bring light fields out of its current confines in pure research, as this work suggests that adding additional layers of displays can augment the depth range and observer accommodation, thus making the whole experience more comfortable, natural, and less nausea-inducing for the user.

### Movie post-production

One of the other big field of application where light fields can make an impact is in post-production, as introduced by Ziegler et al. [54]. Among the first systems designed specifically for this purpose was the Lytro Cinema Camera, released in 2016, a blown up version of the Lytro Illum, featuring very high resolution and framerate, while providing a number of post-processing tools to assist with editing, storage and general workflow. It promised easy refocusing on the fly, and the end of green screens in movie production sets, since in a light field objects could now be extracted based on their depth rather than by using colour information. However, two years later Lytro and this product were abruptly discontinued. While their solution failed to find commercial success, the methods and interest

it created left a lasting mark.

One of the major projects created for a similar purpose is called SAUCE, which stands for Smart Assets for re-Use in Creative Environments. It is a EU project combining the expertise of 8 companies and research groups for creating tools and software for the creative industry. Among those was a light field camera rig which was used for a number of shoots [45], previously mentioned in the section about camera arrays (section 2.2). It consists of an array of 64 full-HD cameras, adjustable in a variety of configurations and distances. By directly capturing a scene using a light field and without manual intervention, it is possible in post-production to apply a number of visual effects, such as changing the focus of the scene for cinematic effect [14]. More broadly, in this context light fields allow more options for a professional director to simulate physically possible, or even impossible, lenses [14].

Other benefits of using light field data include the creation of depth maps, and potentially full geometric reconstructions of the scene. A number of common movie post-production edits such as colour grading or keying could become easier to perform, or, as mentioned before, done using depth information to separate between foreground and background elements [14] and remove the reliance on green screens.

In an attempt to explore the benefits offered by light field data for post-production, in this thesis we look at two applications that become easier to perform and show the potential of the concept. These applications are colour editing and depth-based object separation, and are presented in chapter 4.

### Assisting other fields of research

The Raytrix company still produces 3D light field cameras for industrial use. Some of the available uses include object inspection, with application in self-driving cars, facial recognition, or gesture recognition, or even plant-research and life science. In these cases the main factor is the computation of an accurate depth map which provide enough additional information to perform each task with much higher accuracy compared to using simple 2D cameras.

Additionally these cameras and solutions find use in robotics, with applications in assisted surgery, automated sculpture or package handling. Finally they provide solutions for light microscopy, which allows to capture 3D images of biological structures to better understand them and the way they interact with one another.

## 2.4 Emerging technologies

During the year 2020, the fields of computer vision and computer graphics were shaken by a small revolution, Neural Radiance Fields (NeRF). It was presented in the seminal paper by Mildenhall et al. [18] and allowed as its main selling point to render high quality novel views of a static scene, using only a handful of input images of the scene. This work is the culmination of research on "unstructured light fields" which had been studied by Buehler et al. [55] and Davis et al. [56].

To achieve such results, Mildenhall et al. used a multi-layer perceptron to understand the underlying scene. As opposed to a discrete voxel-grid representation, this allows to represent the scene using a continuous function, and to describe it not only for a particular 3D position but also from any specific viewpoint. Using this representation and classic volume rendering methods, they could easily extract novel views at any position within the range of input image positions, complete with accurate specularity depending on the angle of view, as well as the corresponding high quality disparity map. Both the novel view quality and respective disparity map were of a much higher quality than all the previous state of the art. As another advantage, the input views did not necessitate calibration to be done directly by the user, as it could be obtained automatically using an off-the-shelf structure from motion method [57]. Anyone with a camera could capture a few dozen images of their scene of choice, and feed it to the NeRF network to obtain any number of novel views.

This initial presentation generated a large number of contributions in the field, some only looking at increasing the speed of NeRF, whether training [58, 59], or inference [60, 61, 62, 63, 64]. Others looked at generating deformable NeRFs by allowing to generate novel views of dynamic scenes

with complex non-rigid geometries, in other terms objects moving during capture, as opposed to static scenes only [65, 66, 67, 68, 69]. This allows for instance to add visual effects such as Dolly zoom effects to selfies [65], or to create animated avatars of one's face displaying different expressions [69]. In a similar vein other work looked at generating novel views of captured videos [70, 71, 72, 73], in which the input is a monocular video of the scene. By interpolating either between views, or along the time dimension, or even using both, the resulting novel videos exhibit new characteristics and allow for a number of visual effects.

Other research looked at the type of information NeRF was capable of learning, and tried to expand on this. One such subgroup of research was dedicated to performing shape, reflectance and illumination decompositions [74, 75, 76, 77]. This is an important application which allows to perform relighting of the scene, or colour editing of certain objects, while maintaining realistic output. Others perform the colour editing directly, while still maintaining high fidelity to the input images [78, 79]. Some looked at using NeRF information for the task of pose estimation [80, 81, 82]. This involves taking an already computed NeRF render, and converging toward the pose to estimate by generating a series of guided novel views and giving them a proximity score.

More exotic, some research has been done on compositionality [83, 84, 85] which consists of extracting individual objects from the initial rendered scene, and creating new scenes by moving those objects around, or transforming their size or shape. Finally some are redefining the way texturing is done [86, 87]. Instead of looking at surfaces, like most previous solutions perform, they generate areas of fuzzy geometry around the object, which allows the artist to generate an array of available volumetric textures such as fur, grass, or any kind of fiber, all with adaptable length.

In the last year or so, a wide number of classical computed vision or computer graphics tasks have been revisited using the new power offered by NeRF, and while this new technology and all the new research it spawned was very exciting, our aim was still the development of solutions for aiding high quality image processing and editing using light field information. To evaluate the advantages brought by NeRF to this idea, we performed a

comparative study between traditional state of the art light field methods and NeRF for the applications of novel view synthesis and depth estimation. In addition we looked at the different type of light field data available for use (from camera arrays, gantries, plenoptic cameras, or even synthetic light fields) to see which of the two methods was preferable when working with them. The details of the study will be presented in chapter 5 of this thesis.

## 2.5 Summary

In this chapter, we have given an overview of light fields, from the very theoretical and nearly philosophical initial descriptions in ancient times, to the more practical modern approaches. After having detailed some of the more popular methods of capturing light fields, as well as looking at the different uses they offer, we explored the potential applications of current light field technologies in the real world, outside of research labs. Finally we touched on new emerging methods which no doubt will soon transform how light fields are processed and used, with potential for a larger number of applications. However exciting all of these methods are, most of them still suffer from a number of drawbacks, whether in the need for specific calibrations, very high computing costs, or simply, by nature of their design, visual flaws that make their output difficult to use to its full extent. In the next chapter, we look at one such problem, where the array of images obtained using Lytro cameras has obvious inconsistencies in terms of brightness colour, and generally suffer from different types of noise. We then present a series of software solutions to attenuate or negate these issues, in the hope to increase the usability of such data.

# Chapter 3

# Lytro Image Enhancement

In this chapter we look in further detail at Lytro images, as extracted from RAW data using software applications. These sub-aperture views generally suffer from a number of alterations, which we seek to amend and provide a detailed report of these methods. We show that our methods outperform the previous state of the art extraction toolbox using a number of objective and subjective studies, and highlight the effect of those improvements on a number of typical light field applications.

## 3.1 Introduction

Plenoptic cameras have gained interest in recent years, after the release of two models by the Lytro [13] company which aimed at allowing professionals and amateurs alike to capture light fields with a dense angular sampling, where each micro-lens outputs unfocused micro-images. Unfocused cameras are generally exploited by extracting sub-aperture images (SAI), each with a very wide depth of field and representing different viewpoints of the scene. Shortly after, another similar type of camera called plenoptic 2.0 was developed by Lumsdaine et al. [47]. As opposed to the previous design, here each micro-lens outputs a focused micro-image. Focused cameras are typically used to render focused images where the focus can be dynamically adjusted based on user input. Due to their unique design, plenoptic cameras generate much more complex RAW data compared to traditional cameras, and the exploitation of this data is made more difficult as a result.

Classically, computer vision applications using light field data prefer to use output in the form of SAIs, as they are more practical to handle. In this chapter, we focus our attention on these and explain our method to extract SAIs from unfocused plenoptic camera RAW data. Despite the different solutions proposed by Cho et al. [1], Xu et al. [2] or Seifi et al. [3], the light field toolbox presented by Dansereau et al. [5] is the most widely used in the research community as it offers the most complete pipeline to extract SAIs. It has for instance played a central role in the standardisation effort for light field compression as it is now used as part of the JPEG PLENO [88] test set. The extraction method comprises four steps which can be summarised as follows: a devignetting step first compensates for the vignetting effect of the micro-lenses, i.e. darker pixels on the edges of each micro-lens; demosaicing is then applied to retrieve the RGB colour components of each pixel from the partial colour information actually captured by camera sensors; a compensation of possible rotation of the micro-lens array is performed; finally the pixels are rearranged to compensate for the non-rectangularly aligned hexagonal micro-lenses, in order to convert the image into a set of sub-aperture images.

However, the extracted views suffer from several types of artefacts such

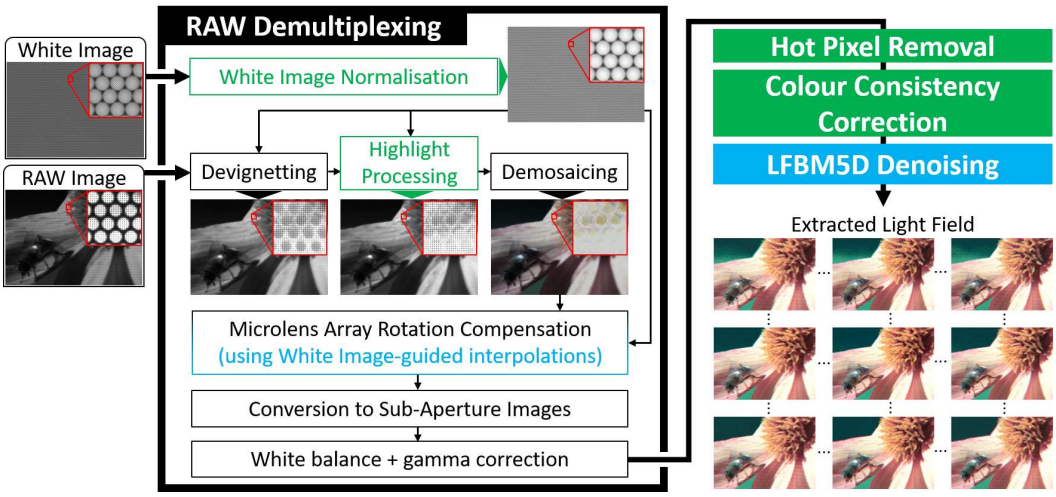



Figure 3.1: Overview of the proposed Light Field Pipeline. The steps in green correspond to the contributions described in this chapter. The steps in blue are state of the art methods that we additionally included in our pipeline, but which are not present in the traditional pipeline [5].

as noise, unnatural horizontal stripes, ghosting effects, colour and brightness inconsistencies on external SAIs, inaccurate colour balance, and substantial loss of dynamic range. Unfortunately, these defects have a negative impact on many light field applications including depth estimation, segmentation, rendering and compression. An overview of these issues is described by Wu et al. [89]. Because of these distortions, a good portion of the external views are generally ignored for these applications. This impacts their results, as using less SAIs means potentially missing out on the critical information they could provide. Although the proprietary Lytro Desktop software compensates for many of these issues, it is still unsuitable for the generation of SAI arrays as its main goal is to render refocused images. Additionally, the software is not officially available anymore, and as a result many users need to use other software solutions to obtain usable images, such as the one presented here.

In this chapter, we propose an improved processing pipeline for lenslet-based plenoptic cameras. First, in order to take the most advantage of the captured RAW data, we propose several improvements on the low level steps of the traditional demultiplexing method of Dansereau et al. [5], converting the RAW data into sub-aperture images. We specifically show that the previously used devignetting step had a negative impact on the

image quality, as it tampered with the colour balance and brightness and caused loss of dynamic range, and we propose ways to correct this.

Additionally, we propose a highlight processing method to compensate for colour issues related to sensor saturation. In order to reduce the ghosting effect of external SAIs, we recommend the use of White Image-guided interpolation following the work of David et al. [90]. Once the sub-aperture images are extracted, further quality enhancement steps are then proposed as post-processing tools. These include hot pixel removal, correction of colour inconsistencies between SAIs and denoising.

Finally, we show the benefits of the different steps of the proposed pipeline by conducting a subjective experiment and analysing the impact of our results on several applications, in comparison with the state of the art demultiplexing proposed by Dansereau et al. The applications studied include light field rendering, compression, super-resolution and editing. We also show that our post-processing step for colour consistency also has practical interest for light fields captured with devices other than plenoptic cameras (e.g. camera array, gantry).

The purpose of this work was to provide a comprehensive framework to process plenoptic data from the RAW output of the camera all the way through various post-processing stages. For clarity, please note that the sections on demultiplexing (Section 3.4) and denoising (Section 3.7) stem from research performed by other members of my team, have been published on their own, and were included in this document only for the purpose of completion. My contributions in this regard are twofold, first bringing the previously mentioned tools together with my solutions detailed in the rest of this chapter (hot pixel removal, colour correction) into a single toolbox. Secondly, performing a thorough objective and subjective comparison of our resulting toolbox with previous state of the art, and its impact of various light field applications.

## 3.2 Related work

### 3.2.1 Demosaicing

In plenoptic cameras, the micro-lens array forms a specific pattern on the sensor, which introduces new difficulties when processing the RAW data. While the light field toolbox presented by Dansereau et al. [5] is capable of converting the RAW data into a set of SAIs, the final images suffer from various artefacts. Further research on the subject has essentially focused on adapting the demosaicing step. For instance, a specific demosaicing method was designed by Yu et al. [91] for focused plenoptic cameras, i.e. plenoptic 2.0. For the more common case of unfocused plenoptic cameras, different optimisation methods have been employed in the demosaicing of Xu et al. [2], Huang et al.[92] and Lian et al.[93]. These methods perform respectively 4D kernel regression, dictionary learning with sparse optimisation, and total variation minimisation. An original approach is proposed by Seifi et al. [3], where the demosaicing is performed after the demultiplexing so that a disparity map can be estimated first, and then used to guide the demosaicing step. Finally, White Image-guided demosaicing and interpolation tools are proposed by David et al. [90] to avoid mixing colour information from different micro-lenses.

However, we believe that a more global analysis of the demultiplexing is necessary, since many inaccuracies can occur in other steps of the pipeline, or during the capture process itself.

### 3.2.2 Devignetting

Dansereau et al. [5] perform lenslet devignetting first as it results in more uniform brightness over the sensor array and thus, easier demosaicing. This step simply consists of a pixel-wise division of the RAW image by a RAW White Image (WI) that exhibits the pattern of micro-lens vignetting. Note that the WI was previously captured during a calibration step by the same device as the picture being processed. However, the red, green and blue filters in the Bayer filter array have different responses to the white light. For this reason, the Bayer pattern is visible on the WI as shown in Figure 3.2(a). Therefore, performing the devignetting step using the un-

processed WI, as in the method of Dansereau et al., interferes with the white balance of the final result.

### 3.2.3 Colour Consistency Correction

After RAW demultiplexing, large differences in colour still exist between the centre and external SAIs, as can be seen in Figure 3.7(b). We refer the reader to the appendix providing insights on how the colour consistency is affected by the demosaicing and its interaction with the devignetting step. To correct this, we chose a recent image recolouring approach proposed by Grogan et al. [94] (described in Section 3.6.1) and adapt it to light fields. Similar to other colour correction approaches proposed in multi-view geometry and panorama stitching applications, such as the ones by Oliveira et al. [95], Park et al. [96], Xia et al. [97] and Hwang et al. [98], this approach uses colour correspondences between a target and palette image to compute a transfer function that maps the colours from the target image to match those of the palette.

### 3.2.4 Highlight Processing

Due to the different saturation levels of the red, green and blue pixels on the sensor, the highlights have unnatural colours. This is a common problem in digital imaging, observed in over-exposed regions after applying the white balance. However, in conventional cameras those regions are typically uniform, making it possible to correct the highlights after the demosaicing (e.g. [99, 100, 101, 102]). In plenoptic cameras, the micro-lens vignetting as well as possible inaccuracies in the devignetting (e.g. slight mismatch between white image and RAW image) and demosaicing steps may create artefacts in those regions for some of the extracted SAIs (see Figure 3.3(b)). The simplest approach for solving the issue is to clip the highlights after the white balance as done by Dansereau et al. [5]. However, this results in a loss of details in the highlights (see Figure 3.3(a)).

### 3.2.5 Denoising

A trivial approach to light field denoising consists of applying an existing single image denoising filter (see Dabov et al. [103], Shao et al. [104] or Jain

et al. [105]) independently to the SAIs. However, better performances are obtained when taking into account the pixel correlation in-between the SAIs. SAIs can, for instance, be stacked in a pseudo-video sequence and denoised using a state of the art video denoiser such as the VBM4D of Maggioni et al. [106]. The angular correlation can also be exploited along the epipolar plane images (EPI): Li et al. [107] use a two-step method which first denoises EPIs taken along a given spatial and angular dimension (e.g. horizontally), and then processes this first estimate using the complementary EPIs (e.g. vertically). Sepas-Moghaddam et al. [108] stack the EPIs in a pseudo-video sequence and denoise using the video denoiser of Maggioni et al. However, none of these methods fully takes advantage of the 4D structure of the light field. Recent improvements in light field denoising performance are thus based on a better exploitation of the 2D angular dimensions. Chen et al. [109] use two joint convolutional neural networks to denoise the light field along the angular and spatial dimensions respectively. Liu et al. [110] denoise light field 4D patches first using a tensor decomposition. The SAIs are then combined into a single high resolution image which is further denoised, and finally projected back into denoised SAIs at the original resolution.

## 3.3 Overview of the Proposed Pipeline

The essential steps of our pipeline are depicted in Figure 3.1. The input data consists of the RAW image formed on the plenoptic camera sensor. Due to the Bayer filter array placed on the sensor, each pixel contains colour information only for one of the RGB components. Another RAW image, called White Image (WI) is obtained by a preliminary calibration process involving the capture of a uniform white surface.

First, a RAW demultiplexing method building upon that of Dansereau et al. [5] is proposed. After a normalisation step, the White Image is used to remove the vignetting in the input RAW image. A novel highlight processing step is then proposed to retrieve natural colours in bright areas where some pixels reach the sensor's saturation level. A standard demosaicing method (Malvar et al. [111]) then recovers the full RGB colour components at each pixel. Similarly to the work of Dansereau et al., we compensate for

slight misalignments between the microlens array and the sensor. The recent White Image-guided interpolation method of David et al. [90] is used for that purpose. The last steps in the method of Dansereau et al. are applied without modification. Pixels are reorganised to convert the lenslet image into a set of sub-aperture images. Due to the hexagonal lenslet pattern, this step includes a resampling of each image from a hexagonal to a square grid of pixels. Finally, white balance and gamma correction are performed. The novel aspects of the RAW demultiplexing and the challenges addressed are presented in Section 3.4, as well as highlight processing in Section 3.4.2).

After the RAW demultiplexing, several defects remain to be corrected. The failure of isolated pixels is a common problem in digital imaging. We choose to correct these so-called 'hot pixels' in a post-processing stage detailed in Section 3.5. A colour correction method is then proposed in Section 3.6 to ensure colour consistency between the light field views. Finally, plenoptic imaging is prone to noise that we remove using the LFBM5D method of Alain et al. [112] (see Section 3.7).

Furthermore, we present a complete evaluation of the pipeline with a subjective study (Section 3.8) and a study of the effect of our quality enhancement tools on various applications (Section 3.9).

## 3.4 RAW Light Field Demultiplexing

### 3.4.1 White Image Normalisation

We correct the White Image (WI) related issue mentioned in sub-section 3.2.2 by multiplying the red and blue pixels of the WI by normalisation factors provided as metadata of the camera and accounting for the different responses of the RGB filters. Note that these factors may also be obtained by colour calibration of the sensor.

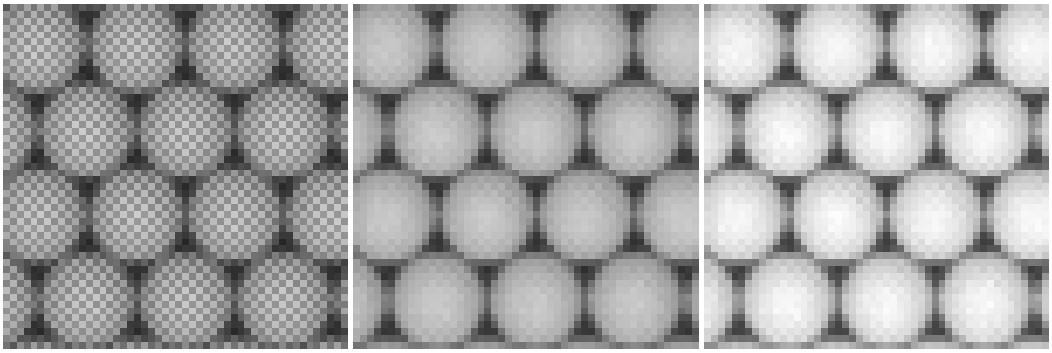
Furthermore, since the pixel values of the WI are lower than $1.0$ even at micro-lens centres, the devignetting of Dansereau et al. also increases the overall brightness of the light field. Bright areas reaching higher values than $1.0$ after devignetting are considered saturated in the rest of the

process, and the information is lost. Therefore, we also apply a global normalisation of the WI by dividing all the pixels by its $99.9^{\text{th}}$ percentile (we do not use the maximum value to exclude possible hot pixels). The effect of the White Image normalisation step on the colours and brightness of the final result is clearly visible in Figure 3.3. However, by decreasing the overall brightness, this normalisation step also reveals unnatural colours in the highlights (see the pink colour in Figure 3.3(b)). We correct this issue in a highlight processing step presented in the next subsection.

### 3.4.2 Highlight Processing

To counter the loss of detail in the highlights resulting from the method of Dansereau et al. [5] described in sub-section 3.2.4, we propose a highlight processing step taking into account the vignetting pattern (i.e. the normalised White Image) and applied before the demosaicing in order to retain the details in the highlights without introducing colour artefacts.

For this step, blocks of four pixels on the RAW image are processed independently. Since the highlight processing is performed before demosaicing, each of the four pixels is associated with only one RGB component organised according to the Bayer pattern. We note the values of these pixels $x_r$, $x_{g_1}$, $x_{g_2}$, $x_b$. Corresponding values in the normalised WI are noted $w_r$, $w_{g_1}$, $w_{g_2}$, $w_b$. In this step, we also take into account the white balance parameters $s_r$, $s_g$, and $s_b$ by which the red, green and blue components will be respectively multiplied later in the white balance step (see Figure 3.1).



(a) (b) (b)

Figure 3.2: Detail of a White Image: (a) unprocessed, (b) after colour normalisation, (c) after both colour and global normalisation.

(a) (b) (c)

Figure 3.3: One view of the light field *duck*: (a) without WI normalisation [5], (b) with WI normalisation, (c) with WI normalisation and highlight processing.

These values can also be interpreted as the saturation levels of each component.

First, we consider saturated pixels such that $x_c \cdot w_c > T$, with $c \in \{r, g_1, g_2, b\}$ and $T$ is a threshold set to $0.99$. Note that $x_c \cdot w_c$ is the original pixel value on the sensor before the devignetting.

Two cases are considered. In the case where the four pixels are saturated, no colour information is present. However, the white balance, applied to those pixels later in the process, results in an unnatural colour. Hence, we cancel the effect of the white balance by setting each pixel of index $c$ to the value $x_c \cdot \hat{s}/s_c$, where $\hat{s} = \max(s_r, s_g, s_b)$. When at least one of the four pixels is not saturated, we find the index $m$ of the pixel with lowest value. A saturated pixel of index $c$ then takes the value $x_m \cdot s_m/s_c$. However, in practice, separating these two cases may cause abrupt changes of brightness. Therefore, we blend between these two behaviours using the following formula for modifying a saturated pixel $x_c$ into $x'_c$:

$$x'_c = \max\left((1-\alpha)\frac{x_m \cdot s_m}{s_c} + \alpha\frac{x_c \cdot \hat{s}}{s_c}, x_c\right), \tag{3.1}$$

where $\alpha \in [0, 1]$ is the blending parameter indicating the total amount of saturation as $\alpha = \min(1, x_m \cdot \frac{1}{4}\sum_c w_c)^2$. The maximum between the modified and the original value is used since $x_m \cdot s_m/s_c$ may be lower than the original saturated pixel $x_c$. This operation prevents possible discontinuities with neighbour pixels slightly below the saturation detection thresh-

old.

Note that after the white balance step, the regions recovered by the highlight processing may have values above 1. In order to retain those details in the final image without affecting the overall brightness, we apply a soft saturation function $softSat$ to each pixel after the white balance step:

$$softSat(x) = 1 - \frac{\ln(1 + e^{R(1-x)})}{\ln(1 + e^{R})}, \tag{3.2}$$

where $R$ is a parameter controlling the smoothness of the curve (lower $R$ resulting in smoother saturation). We set $R = 7$ in our implementation. The soft saturation curve is illustrated in Figure 3.4 and the final result is shown in Figure 3.3(c).

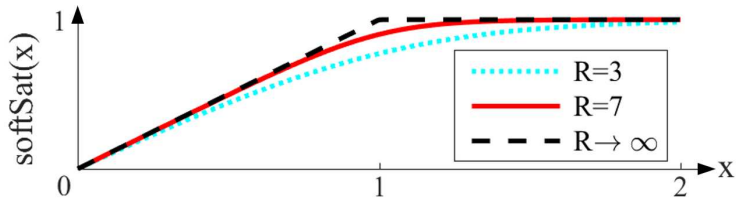



Figure 3.4: Soft saturation function with different parameters $R$.

### 3.4.3 White Image-guided Interpolations

Previous analysis by David et al. [90] has shown how standard demosaicing and interpolations introduced both ghosting artefacts and fading of the colours in the external SAIs. In order to reduce the problem, they

(a) (b) (c)

Figure 3.5: Advantages and limitations of the White Image-guided method of [90]: (a) standard demosaicing [111] and bicubic interpolations, (b) standard demosaicing [111] and WI-guided interpolations, (c) WI-guided demosaicing and interpolations.

adapted those steps by weighting the contribution of each pixel using the vignetting pattern of the White Image. Two observations can be made from their results. Firstly, the ghosting effect is essentially reduced by the adaptation of the interpolation step (see Figure 3.5(b)). Secondly, while their modified demosaicing improves the overall colour consistency between SAIs, it may also create colour noise (see Figure 3.5(c)). Hence, we suggest that only the WI-guided interpolations should be used, and we propose in Section 3.6 a post-processing step to enforce colour homogeneity in the light field.

## 3.5 Hot Pixel Removal

Hot pixels are isolated pixels taking extreme values due to internal errors on the camera sensor. Their detection within the RAW demultiplexing stage is challenging due to the fact that the demosaicing step retrieves inaccurate colours, not only for the hot pixels, but also for their neighbours, corresponding to angular neighbours in the light field. However, in the sub-aperture images obtained by the demultiplexing, the spatial neighbours of the hot pixels are unaffected. Furthermore, hot pixels are not accurately removed by traditional light field denoising methods, such as the ones presented in Section 3.7. Thus we directly perform hot pixel removal after RAW demultiplexing.

A typical issue for hot pixels is the fact that they exhibit extreme values in their colour components, but this in itself is not a sufficient criteria for detection. Instead, for each SAI $I$, we identify hot pixels by comparing the colour values $x^i$ of each pixel $i$ to those of its neighbours in $\Omega_{n\times n}(i)$, the $n\times n$ window centred on the pixel $i$. Based on this, we compute a probabilistic measure $\rho^i$ to indicate how likely $i$ is to be a hot pixel and threshold this value to detect the most likely hot pixels in the SAI. We tested colour values in both the RGB and CIELAB colour spaces [113], and found that CIELAB helped us identify hot pixels more easily, so we chose to use this colour space exclusively. We use Matlab's *rgb2lab* function to convert images to CIELAB d65, with L* taking values between $0$ and $100$ and a* and b* values between $\pm 110$.

The procedure we use to detect hot pixels is described in Algorithm 1. For each pixel colour $x^i$ in CIELAB space, if it lies within a colour distance $t_d$ to only a small number of pixels (less than $t_c$) in the window $\Omega_{n\times n}(i)$, the value $\rho^i$ will be high (see Algorithm 1). The distance we use in CIELAB space is the Euclidean distance. In Figure 3.6(b) we display $\rho^i$ values for each pixel in SAI (a), with the red values in (b) showing pixels with the highest $\rho$ value. We then threshold these values in order to detect the most likely hot pixels, with $\rho^i > t_\rho$ selected (Figure 3.6(c)). Since hot pixels do not typically appear as white in an image, we also add a check to make sure pixels that lie within a distance $t_w$ of the colour white (such as small regions of white highlights on an object) are not incorrectly detected as hot pixels (see Algo. 1). Here, we can see that our detection method is robust to colour changes along edges, with very few edges being detected incorrectly as hot pixels. Finally, we correct the hot pixel $i$ using a $3\times 3$ median filter centred on it, which takes the median value for the L*, a* and b* components (ignoring the hot pixel values) and applies it to the hot pixel. Edge and corner pixels and processed by duplicating the external lines of pixels before running the filter. While this runs the risk of duplicating hot pixels and inducing artefacts in theory, our practical application found no example of this occurring. Figure 3.6(d) shows the final results in which the isolated red and green hot pixels have been successfully detected and restored via our hot pixel removal.

## 3.6 Colour Consistency Correction

In [94], Grogan et al. show that their approach outperforms several leading colour correction approaches [114, 115, 116, 117, 97, 96] when applied to images with similar content. Overall, their correspondence based method is shown to outperform those that do not consider correspondences [114, 115, 116] while their flexible thin plate spline colour transfer function allows them to correct more non-linear colour differences between images, outperforming methods whose transfer functions depend on only a small number of parameters [96, 97]. They also found that Hwang et al's method [117] can introduce visual artefacts when correspondence outliers are used to estimate the transfer function, while Grogan et al.'s cost function is

**Result:** SAI $I$ with hot pixels removed.
Define thresholds $t_d = 30$, $t_w = 30$ and $t_\rho = 0.8$, window size $n = 7$ ;
**for** $i \in I$ **do**
  Compute $\Omega_{n\times n}(i)$;
  `/* Compute hot pixels probability map */`
  Define $count = 0$;
  **for** $i' \in \Omega_{n\times n}(i)$ **do**
    **if** $||x^{i'} - x^i||_2 < t_d$ **then**
      $count \leftarrow count + 1$;
    **end**
  **end**
  $\rho^i = 1 - \frac{count}{n^2}$;
  `/* Filter hot pixels */`
  **if** *($\rho^i > t_\rho$ and $||white - x^i||_2 > t_w$)* **then**
    $x^i \leftarrow \text{median}_{L^*a^*b^*}(\Omega_{3\times 3}(i) - \{i\})$ ;
  **end**
**end**

**Algorithm 1:** The process used to detect and correct hot pixels in an SAI $I$. Here, $white = [100, 0, 0]$ is the colour white in CIELAB space and $||\cdot||_2$ denotes the Euclidean distance.

shown to be more robust to outlier pairs, with the smooth transfer function also ensuring that similar colours stay similar after recolouring. For these reasons, we decided to adapt Grogan et al's method to light field data, and in this section give further details about our approach.

### 3.6.1 Correspondence Estimation

For the colour transfer algorithm to produce good results, we needed to compute accurate correspondences between both views. We explored existing methods for correspondence estimation between SAIs following the example of Chen et al. [9], who used optical flow successfully in their work on light fields and chose to adapt a similar method. As the colour transfer algorithm does not require that all the pixels of an image pair are matched to obtain satisfying results, a preference towards lower computational complexity was taken in this step. We therefore used only the first step of coarse-to-fine patch matching (CPM) developed by Hu et al. [118] to obtain a set of sparse correspondences between pairs of views.

It is similar to PatchMatch [119] and works by taking $n$ pixels on a regu-

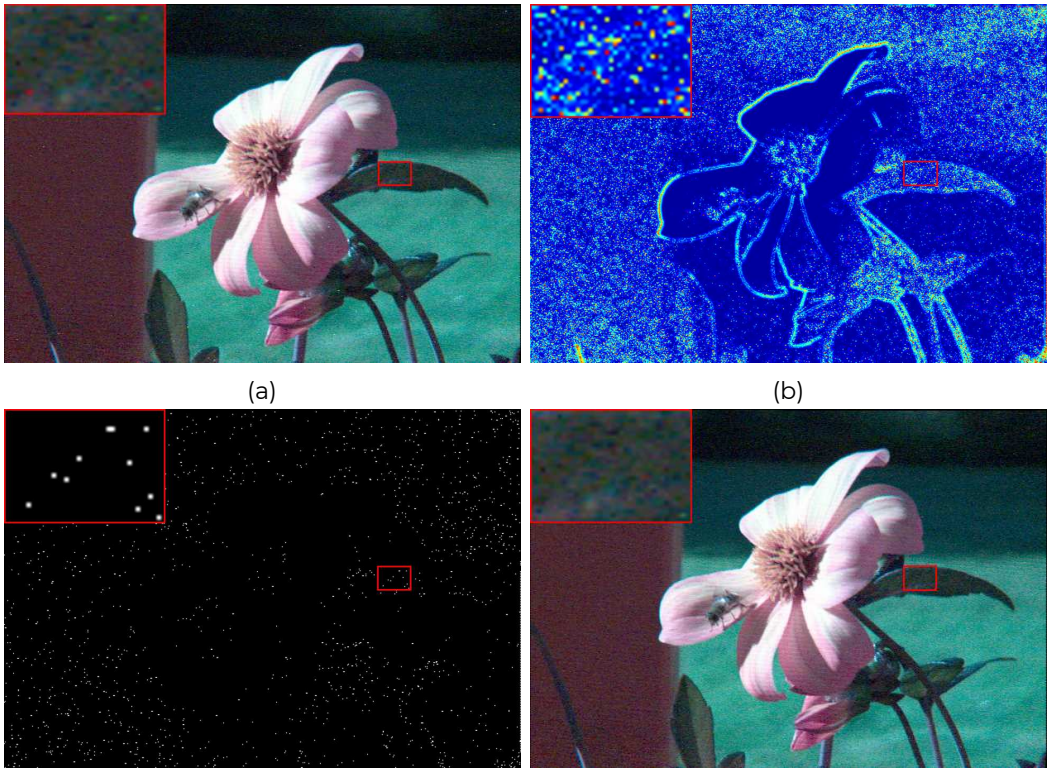



Figure 3.6: (a) Input SAI with zoom clearly showing red and green hot pixels as described in Section 3.5; (b) heat map showing values $\rho^i$ for all pixels $i$; (c) detected hot pixels with $\rho^i > t_\rho$; (d) our corrected SAI.

lar grid in the target SAI as seed pixels, noted $c_t^{(n)}$, and finds their matching pixels, or correspondences, in the palette view, noted $c_p^{(n)}$. To compute these correspondences, a candidate set of correspondences is first found using SIFT features. In the second step, points are sampled around each candidate correspondence, and if they prove to be more accurate, replace the original. This process iterates a number of times until a globally stable set of correspondences is found. Finally, outliers are detected and removed from the pool, creating the final set of correspondences $\{c_t^{(n)}, c_p^{(n)}\}$.

This process provides us with on average $30$k correspondences between two images of size $625 \times 434$. We extract the colour information of these pixel pairs to form the colour models representing both target and palette images and pass them on to the colour transfer algorithm.

(a) RAW demultiplexing by Dansereau et al.

(b) Proposed RAW demultiplexing

(c) Proposed Recolouring step

Figure 3.7: Matrix of sub-aperture images of the *bee_2* light field. This view allows to better perceive the improvement on the fidelity of the colours our demultiplexing (b) offers over the demultiplexing of Dansereau et al. [5] (a), and highlights the colour inconsistencies on the external views that we fix (c).

### 3.6.2 Colour Transfer

Given a set of $n$ colour correspondences $(c_t^{(k)}, c_p^{(k)})_{k=1...n}$ between the target and palette image, where the set of colours $c_t^{(k)}$ from the target image should correspond to the colours $c_p^{(k)}$ from the palette after recolouring, Grogan et al. [94] propose to fit a Gaussian Mixture Model to each set of correspondences as follows:

$$p_t(x|\theta) = \sum_{k=1}^{\mathrm{n}} \frac{1}{\mathrm{n}} \mathcal{N}(x;\, \phi(c_t^{(k)}, \theta),\, h^2 \mathrm{I}) \tag{3.3}$$

and

$$p_p(x) = \sum_{k=1}^{\mathrm{n}} \frac{1}{\mathrm{n}} \mathcal{N}(x;\, c_p^{(k)},\, h^2 \mathrm{I}) \tag{3.4}$$

Each Gaussian is associated with an identical isotropic covariance matrix

$h^2\mathrm{I}$, and the vector $x \in \mathbb{R}^3$ represents values from a 3D colour space. Transforming the colours $c_t^{(k)}$ by some transformation $\phi$ which depends on $\theta$ creates the colours $\phi(c_t^{(k)}, \theta)$. The goal is to transform the colour distribution of the target image to match that of the palette image by estimating the transformation $\phi$ that registers $p_t(x|\theta)$ to $p_p(x)$. Grogan et al. propose letting $\phi$ be a global parametric thin plate spline transformation:

$$\phi(x, \theta) = \underbrace{\mathrm{A}\, x + o}_{\text{Affine}} + \underbrace{\sum_{j=1}^{m} -w_j \, \|x - q_j\|_2}_{\text{nonlinear}} \tag{3.5}$$

with $\theta = \{A, o, w_j\}$ the parameters to be estimated. Here, $A$ is an affine matrix, $o$ is a translation offset vector and $\{w_j \in \mathbb{R}^3\}$ are coefficients controlling the non-linear part of the transformation with $\{q_j\}_{j=1,..,m}$ a set of control points evenly sampled in the colour space.

To estimate the parameter $\theta$ controlling $\phi$, the following is minimised:

$$\mathcal{C}(\theta) = -\langle p_t | p_p \rangle = \sum_{k=1}^{n} \frac{1}{n^2} \mathcal{N}(0;\, \phi(c_t^{(k)},\, \theta) - c_p^{(k)},\, 2h^2\mathrm{I}) \tag{3.6}$$

For our application, better results were obtained using the CIELAB colour space rather than the RGB colour space. Similar to [94], we add a regularisation term to ensure our thin plate spline function is smooth. We also found that additional steps had to be taken when optimising this cost function to avoid local minima. Therefore we used a two step process to estimate $\theta$. The first step computes an initial estimate for $\theta$ using a subsample of the correspondences (computed using k-means with $K = 1000$). In the second step, the parameters $A$ and $o$ are fixed and only the non-linear parameters $w_j$ are refined using the full set of correspondences. We found that this two step process ensured local minima were avoided and the correct solution was found.

### 3.6.3 Propagation

As an improvement on our previous work [19] we decide here to focus on the propagation scheme that allows for the best visual quality. Our goal here is to guarantee two things: firstly that colours be consistent across the light field, i.e. two consecutive views should not exhibit any visible difference between them, and secondly that true scene colours be preserved as much as possible in all the views.

The propagation scheme we use in this work is twofold. The demultiplexing step of our pipeline ensures we obtain natural colours in all the views, with the central views displaying the most accurate colours. Therefore, when recolouring a target SAI $T$ in the light field we first compute correspondences between $T$ and the centre view $M$ of the light field using the method described in Section 3.6.1. To ensure $T$ displays similar colours to its neighbouring images, we also compute correspondences between $T$ and its inner neighbouring view $P$. If $T$ lies on the central column of the light field, its inner neighbouring view $P$ also lies on the central column, either above or below $T$ depending on which is closest to the centre view $M$. Otherwise, $P$ will lie on the same row of the light field as $T$, again either to the left or right of $T$ depending on which is closest to the centre view $M$. For each target SAI $T$, this combination of correspondences is then input into Eqs. (3.3) and (3.4), meaning each view will be recoloured using a function computed using correspondences from the centre view and its inner neighbour. We recolour each SAI in the light field starting with the centre column, from the centre view and outward, then in every row, from the middle view outward. This procedure is described in Algorithm 2, with a visual explanation given in Figure 3.8.

The choice to include the previously-recoloured neighbouring views was made empirically. When only using the centre view, artefacts occur due to large parallax between external views and the centre view. On the other hand using only the inner neighbour views can cause a slight fading of colours as we move toward the edges of the light field, as each successive recolouring causes a minor loss of colour intensity. Therefore, the use of two views simultaneously as palette images helps us ensure that we get both the most vivid colour in every view, and a reduction in the possible

artefacts introduced by the method.

**Result:** Colour corrected Light Field with $m \times m$ SAIs.
Define $M = I_{(\lceil \frac{m}{2} \rceil, \lceil \frac{m}{2} \rceil)}$;
**for** $j = 0 : (\lfloor \frac{m}{2} \rfloor - 1)$ **do**
    /* Centre column, downward direction */
    colCorrect($\lceil \frac{m}{2} \rceil + j + 1, \lceil \frac{m}{2} \rceil, \lceil \frac{m}{2} \rceil + j, \lceil \frac{m}{2} \rceil$);
    /* Centre column, upward direction */
    colCorrect($\lceil \frac{m}{2} \rceil - j - 1, \lceil \frac{m}{2} \rceil, \lceil \frac{m}{2} \rceil - j, \lceil \frac{m}{2} \rceil$);
**end**
**for** $k = 0 : \lfloor \frac{m}{2} \rfloor$ **do**
    **for** $j = 0 : (\lfloor \frac{m}{2} \rfloor - 1)$ **do**
        /* every row, from centre SAI to right */
        colCorrect($\lceil \frac{m}{2} \rceil \pm k, \lceil \frac{m}{2} \rceil + j + 1, \lceil \frac{m}{2} \rceil \pm k, \lceil \frac{m}{2} \rceil + j$));
        /* every row, from centre SAI to left */
        colCorrect($\lceil \frac{m}{2} \rceil \pm k, \lceil \frac{m}{2} \rceil - j - 1, \lceil \frac{m}{2} \rceil \pm k, \lceil \frac{m}{2} \rceil - j$);
    **end**
**end**

**Function** colCorrect($row_T$, $col_T$, $row_P$, $col_P$)**:**
    $T = I_{(row_T, col_T)}$;
    $P = I_{(row_P, col_P)}$;
    $(c_t, c_p) = (c_T, c_P) \cup (c_T, c_M)$;
    $\hat{\theta} = \text{argmin}_\theta \mathcal{C}(\theta)$;
    $I_{(row_T, col_T)} \leftarrow \phi(T, \hat{\theta})$;
    **return**;

**Algorithm 2:** The propagation technique used to recolour the entire light field. The blue and red regions correspond to the blue and red arrows in Figure 3.8.

## 3.7 Denoising

In addition to the colour artefacts addressed previously, lenslet plenoptic cameras have by design a lower signal to noise ratio than single lens cameras, since light rays coming from different angular directions are no longer averaged on a single pixel sensor. Thus we propose to apply denoising as a final step of the pipeline. In conventional photography, it is sometimes preferred to perform denoising either before or jointly with the demosaicing step when the RAW data is available (e.g. [120, 121, 122]). However, applying such denoising methods on plenoptic RAW data would not

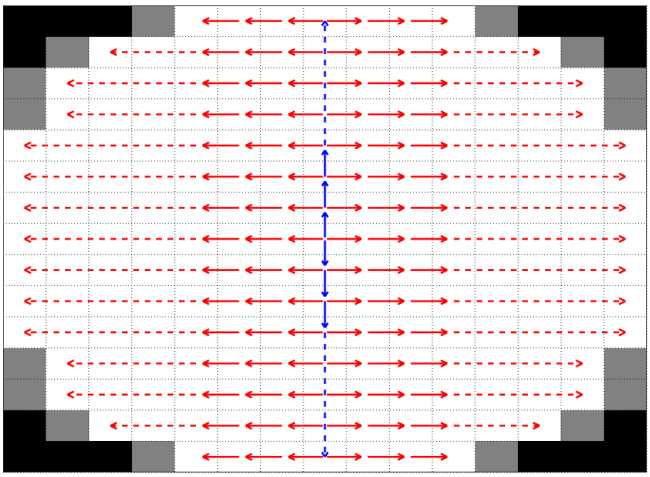


Figure 3.8: Pattern representing our propagation of colours in a light field. The centre column is processed first, then each row.

exploit the redundancies in the 4 dimensions of the light field. Furthermore, neighbour pixels on the sensor may correspond to different lenslets and thus belong to diametrically opposed SAIs. Therefore, applying denoising in the early stages of the pipeline is likely to produce cross-talk artefacts on the external SAIs. A similar issue was observed in Figure 3.5(a) when using linear interpolation for the lenslet array rotation step. Denoising is then preferably applied at the end of the process, after the colour correction step, since the latter helps to improve the consistency of the light field over the angular dimensions. This benefits most existing light field denoising methods, which rely on the angular redundancy.

Any of the denoising methods cited in Section 3.2.5 could be used in the proposed pipeline, but we choose the state of the art LFBM5D filter, which was shown to perform well on lenslet light fields by Alain et al. [112]. The core idea of this filter is to exploit redundancies over the light field angular and spatial dimensions, as well as self-similarities occurring in natural images. As in the BM3D filter of Dabov et al. [103] or the VBM4D filter of Maggioni et al. [106], the LFBM5D filter exploits the non-local self-similarities occurring in natural images, in addition to the spatio-angular redundancies. 5D patches built from similar 2D patches are filtered in the 5D transform domain, where their spectrum is very sparse and offer a good decorrelation between the true underlying signal and noise coefficients. Noise can thus be filtered by applying hard-thresholding on the

5D transform coefficients in a first step, and Wiener filtering in a second step. The LFBM5D output is then obtained by applying the inverse 5D transform on the filtered 5D spectrum.

The denoised light field is the output of the proposed pipeline, and we evaluate the full performance of the pipeline in the next section, as well as its preprocessing advantages for several light field applications in Section 3.9.

## 3.8 Validation of the Proposed Pipeline

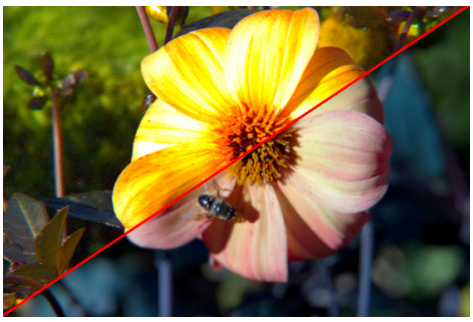
We use a variety of metrics and experiments to validate the effectiveness of our pipeline. 17 light field sets were chosen from the EPFL [123] and INRIA [124] datasets captured with Lytro Illum cameras, as well as datasets captured using our own Lytro Illum camera; those include one set featuring non-Lambertian objects, in order to study the effect of these on selected applications. A metric analysis of 10 light fields from the recent Stanford dataset [125] can also be found in the appendix.

In order to validate the different steps of the proposed pipeline, we consider the following seven combinations of settings (see Table 3.1) : *1-Da)* demultiplexing of Dansereau et al. [5], *2-De)* proposed demultiplexing (Section 3.4), *3-DeH)* proposed demultiplexing + Hot Pixel Removal (HPR) (Section 3.5), *4-Re)* proposed recolouring (Section 3.6), *5-DaN)* toolbox of Dansereau et al. + our denoising (Section 3.7), *6-DeN)* our demultiplexing + our denoising, and *7-ReN)* our full pipeline (demultiplexing, HPR, recolouring, denoising).

Note that other demultiplexing methods have been presented in [1, 2, 3]. However, similarly to [5], they do not consider the issues of wrong white balance and exposure, saturated highlights, colour inconsistencies, hot pixels and noise. Therefore, this section only presents comparisons against the method [5] which we have built upon. Nevertheless, further review and evaluation of the relevant tools in [1, 2, 3] as well as the more recent PlenoptiCam software [4] are given in the appendix.

Table 3.1: Details of the processing applied to the different groups of images or videos used for the validation of the proposed pipeline.

| | $Da$ | $De$ | $DeH$ | $Re$ | $DaN$ | $DeN$ | $ReN$ |
|---|---|---|---|---|---|---|---|
| Dansereau et al. [5] | ✓ | | | | ✓ | | |
| Our demultiplexing (Sec. 3.4) | | ✓ | ✓ | ✓ | | ✓ | ✓ |
| HPR (Sec. 3.5) | | | ✓ | ✓ | | | ✓ |
| Recolouring (Sec. 3.6) | | | | ✓ | | | ✓ |
| Denoising (Sec. 3.7) | | | | | ✓ | ✓ | ✓ |



(a) Dansereau et al. vs Lytro Desktop

(b) Our method vs Lytro Desktop

Figure 3.9: Below red line: refocused image from Lytro Desktop proprietary software (using 'as shot' white balance option). Above red line: central SAI of the *bee_2* light field obtained with (a) Dansereau et al.'s method [5], (b) our method. (Standard sRGB gamma correction is performed in both cases.)

### 3.8.1 Colour Consistency

We first show in Figure 3.9 the importance of the simple normalisation steps proposed in Section 3.4.1 for the colour balance and overall brightness. For reference, the bottom right part of each sub-figure shows a refocused image obtained by the Lytro proprietary software with the intended colours, i.e. as displayed by the camera when taking the picture. Note that the results of Dansereau et al. [5] are often wrongly assumed to be gamma corrected, leading to exaggerated contrasts and colour saturation. For a fair comparison, we performed standard sRGB gamma correction for both methods.

We used several metrics to evaluate the colour accuracy of our processed pipeline results including PSNR, SSIM [126], S-CIELab [127] and a histogram distance metric. For each metric, we use the centre SAI as reference and compute the distance between it and all other SAIs in the light field, and averaged the results over all SAIs. We used the centre view as reference

for these metrics since the colours in the centre view are the most accurate and are not affected by the colour fading artefacts present in the outside views. Disparity differences between the centre view and all other SAIs may affect the evaluation, but since all methods are compared on the same set of light fields with the same disparity differences, metric values are still indicative of colour correction accuracy. PSNR and SSIM were computed per colour channel and averaged. The results can be seen in Figure 3.10. As PSNR, S-CIELab and SSIM capture local colour differences between images, their accuracy can be affected by disparity changes between SAIs. As a result we have also included a global histogram distance which is more robust to changes in the image. For a pair of images, to compute this histogram distance we calculated the average chi-square differences between their L*, a* and b* histograms, each computed on 25 bins.

In Figure 3.10, we compare the colour consistency of results generated with *Da*, *DeH*, *Re* and *ReN*. In terms of PSNR, SSIM, and S-CIELab, *Da* performs the worst in all cases, followed by *DeH*, *Re* and *ReN*, confirming that each step of our pipeline improves the consistency of the light field and its fidelity with the centre SAI. The histogram distance results tell a similar story, with the initial decoding methods *Da* and *DeH* performing the worst in general, followed by *Re* and *ReN*. However, this metric indicates that in some cases, *Da* and *DeH* are more consistent than *Re* and *ReN* (*raoul* and *la_guin*).

Upon close inspection we found that some colour inconsistencies present after decoding (*DeH*) were not successfully removed after recolouring (*Re*) due to the smooth, global nature of our thin plate spline colour transfer function which ensures that similar colours in the image cannot become very different after recolouring. For example, in the *raoul* light field, large portions of the red background were darker in colour in the outside SAIs (see Fig 3.11 (a)). After recolouring using our technique, the dark red regions were brightened to match the centre image, but this also caused the brown colour of the cats fur, which has pixels similar in colour to the dark brown in the background, to become more red (see Fig 3.11 (b)). Therefore, although large portions of the recoloured outside and centre SAIs are similar, other smaller regions may still differ slightly in colour. This

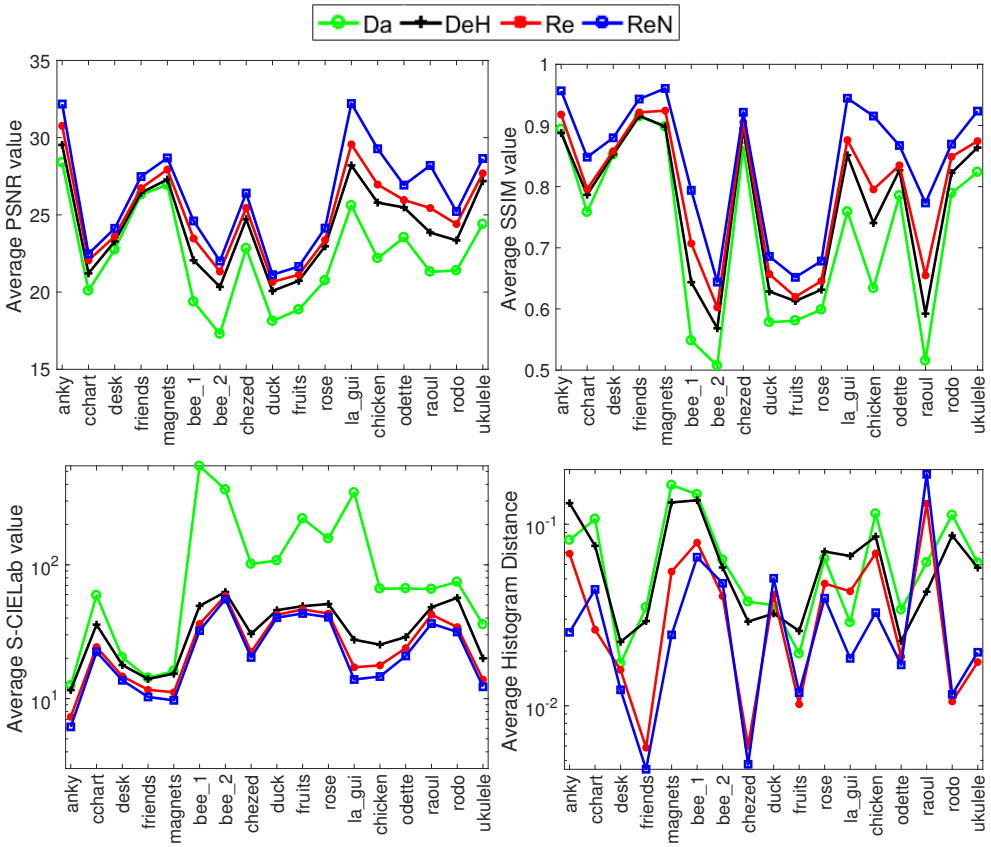



Figure 3.10: Metric comparison, using PSNR, SSIM [126], S-CIELab [127] and histogram distance. Higher values are better in terms of PSNR and SSIM, and lower are better for S-CIELab and the histogram distance.

explains the spike in colour consistency appearing in the local histogram metrics for the *raoul* and *la_guin* light fields. However, we found that these artefacts do not occur regularly, and even when they are present, our propagation technique ensures colours change gradually across the light field SAIs, with neighbouring images displaying similar colours with only slight colour variations. Our subjective experiments also highlight that even in these cases, the recoloured SAIs are more pleasing than those without recolouring (see Table 3.4).

Overall, we see that each step of the proposed pipeline improves colour consistency and reduces the colour or histogram distances while improving the structural similarity by bringing brightness and contrast to similar levels, and overall lowering pixel-wise error.

(a) (b)

Figure 3.11: The centre SAI in *raoul* is overlaid in column blocks onto one of the outside SAIs before recolouring (a) and after recolouring (b). The colours at the bottom of the images indicate which SAIs the columns are taken from - the centre SAI (blue), the outside SAI before recolouring (green) or the outside SAI after recolouring (red). The colour differences in the red background between the centre and outside SAIs in (a) are successfully removed in (b) but slight reddish tones are introduced into the cat's fur.

Figure 3.12: Recolouring examples on the *cchart* and *bee_2* light fields. The first column shows the centre SAI (red and blue lines are used to create the EPIs in Figure 3.13); the second column is one of the external views, notice the apparent washing out of the colours compared to the centre view; the third column is the same view after our recolouring, restoring most of the original colours.

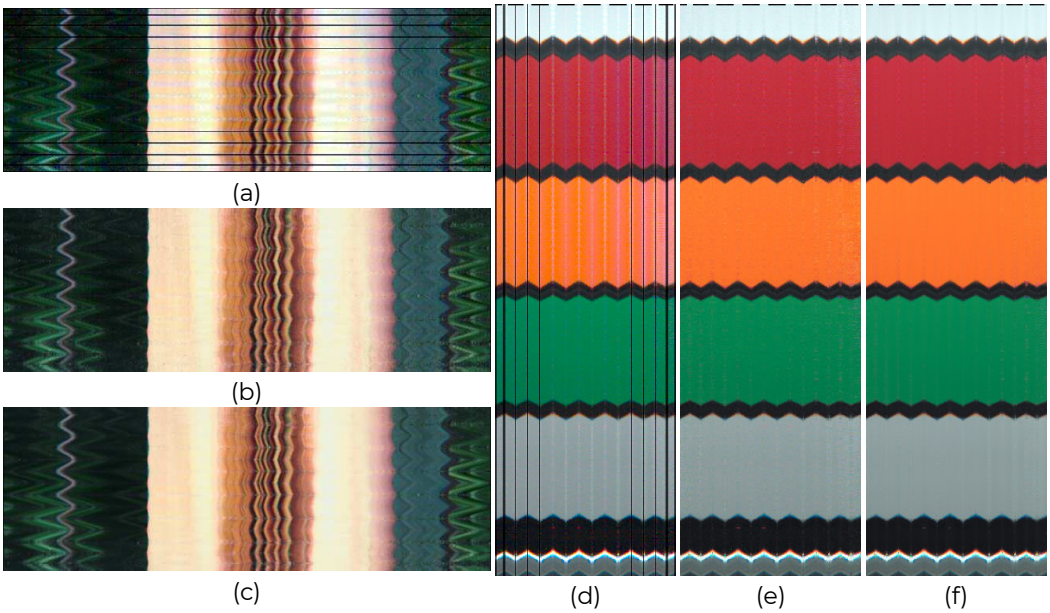



Figure 3.13: Stacked EPIs showcasing colour differences in the *bee_2* (a,b,c) and *cchart* (d,e,f) light fields: after our RAW demultiplexing (a,d), after recolouring (b,e), and after denoising (c,f). Dark lines in (a,d) are caused by the dark SAIs in the corner of the light field (see Figure 3.15) which are corrected by our recolouring. Selected lines are shown in Figure 3.12.

We visually assess the results of our recolouring method in Figures 3.7, 3.12 and 3.13. The results are visually pleasing, with smooth transitions between consecutive views, seen in Figure 3.7, and the colours overall remaining consistent with those in the centre view (see also Figure 3.12). This is particularly visible when computing EPIs (as seen in Figure 3.13), which consist of stacks of the same horizontal or vertical line of pixels taken across all the views of the light field. These images show a clear improvement in colour consistency over the whole light field, which is further improved after the denoising process.

### 3.8.2 Noise Analysis

#### Analysis on a ground truth noise free dataset

Since the light fields captured with the Lytro camera do not have a noise free ground truth, we propose to quantify the noise level by performing blind noise level estimation. For that purpose we use the method of Chen et al. [128], which estimates the noise level of an image based on the eigen-

Table 3.2: Noise level $\sigma_{est}$ estimated using [128] for each light field and each setting combination described in Table 3.1. The 3 setting combinations including denoising are shown on the right.

| $\sigma_{est}$ | $Da$ | $De$ | $DeH$ | $Re$ | $DaN$ | $DeN$ | $ReN$ |
|---|---|---|---|---|---|---|---|
| anky | 2.62 | 1.91 | 1.90 | 1.82 | 0.86 | **0.48** | 0.51 |
| cchart | 1.92 | 1.99 | 1.99 | 1.53 | **0.54** | 0.63 | 0.65 |
| desk | 3.45 | 3.06 | 3.11 | 2.85 | 1.18 | **0.97** | 1.12 |
| friends | 3.18 | 3.02 | 3.03 | 2.93 | 1.96 | **1.91** | 1.96 |
| magnets | 2.86 | 2.85 | 2.85 | 2.86 | **1.87** | 1.89 | 2.02 |
| bee_1 | 2.08 | 2.13 | 2.13 | 1.80 | **0.79** | 0.89 | 0.97 |
| bee_2 | 8.53 | 5.73 | 5.69 | 3.97 | 6.68 | 3.42 | **2.04** |
| chezed | 8.20 | 5.44 | 5.41 | 3.72 | 6.29 | 3.06 | **1.55** |
| duck | 6.60 | 5.40 | 5.49 | 5.74 | 5.76 | **4.66** | 5.05 |
| fruits | 5.87 | 4.36 | 4.39 | 3.48 | 4.45 | 3.11 | **2.50** |
| rose | 5.12 | 3.91 | 3.92 | 3.38 | 3.28 | 2.29 | **2.01** |
| la_guin | 4.07 | 3.06 | 3.06 | 2.47 | 1.64 | 0.88 | **0.71** |
| chicken | 4.90 | 3.29 | 3.32 | 2.86 | 3.23 | 1.80 | **1.69** |
| odette | 5.93 | 4.29 | 4.20 | 2.99 | 3.98 | 2.25 | **1.48** |
| raoul | 3.94 | 3.18 | 3.22 | 3.09 | 2.53 | **2.02** | 2.03 |
| rodo | 8.12 | 6.26 | 6.23 | 3.79 | 6.21 | 4.10 | **1.85** |
| ukulele | 4.42 | 3.43 | 3.44 | 3.45 | 2.94 | **2.14** | 2.24 |
| Average | 4.81 | 3.72 | 3.73 | 3.10 | 3.19 | 2.15 | **1.79** |

values of the covariance matrix of the image patches, based on an Additive White Gaussian Noise (AWGN) model.

To first validate the assumption that the noise of the Lytro camera follows the AWGN model, we created a noisy light field dataset consisting of 5 scenes. For each scene, 3 different noise levels were created by changing the ISO gain and maximising the shutter speed so that the image is as bright as possible without saturation. For each scene and ISO setting, ~30 noisy instances were captured, and a ground truth noise free light field was created by averaging the noisy instances. We ensured that the lighting conditions remained stable. The light field noise can then be obtained by removing the noise free light field from the noisy instance. By analysing the histograms of the light field noise, we observed that the AWGN model is validated for each SAI of the light field. By fitting a normal distribution to the histograms, we then obtained the ground truth noise level for each colour channel as the standard deviation of the normal dis-

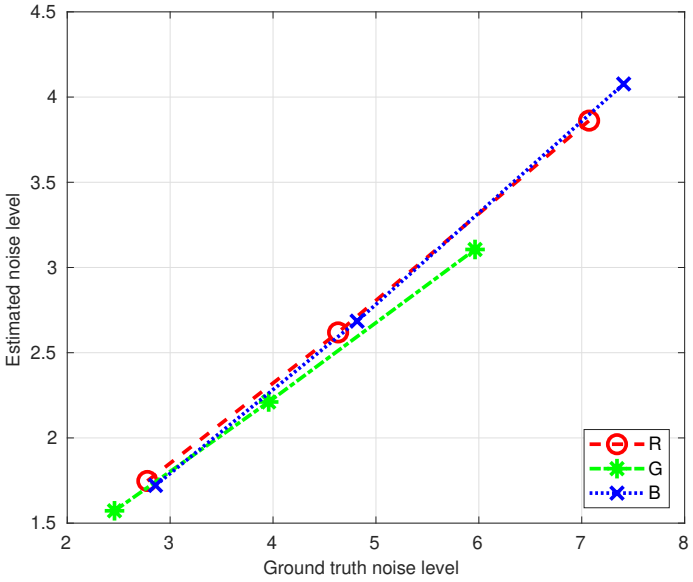



Figure 3.14: Blind noise level estimation [128] plotted against the ground truth noise level, averaged over the 5 light fields of the noisy dataset. Although the no-reference metric from [128] does not estimate the exact noise level, it can be use for relative comparison.

tribution. More details on the dataset are given in the appendix.

Finally, we evaluated the chosen blind metric [128] by comparing the estimated noise level to the ground truth. The graph of Figure 3.14 shows the estimated noise level, averaged over all SAIs and all light fields, against the ground truth noise level. While the blind metric does not evaluate the exact noise level, a near linear relationship between the ground truth and estimated noise level can be observed, which validates the use of the chosen metric for the evaluation of our pipeline.

### Noise level estimation of the proposed pipeline

Here we estimate the noise level after each step of the pipeline using the blind metric [128]. The noise level of the whole light field is computed by first independently estimating the noise level of each SAI, and then averaging the results. Results are shown in Table 3.2 for all setting combinations described in Table 3.1 and all 17 test light fields.

We observe that our proposed demultiplexing method can slightly reduce the noise level compared to Dansereau. The hot pixel removal step

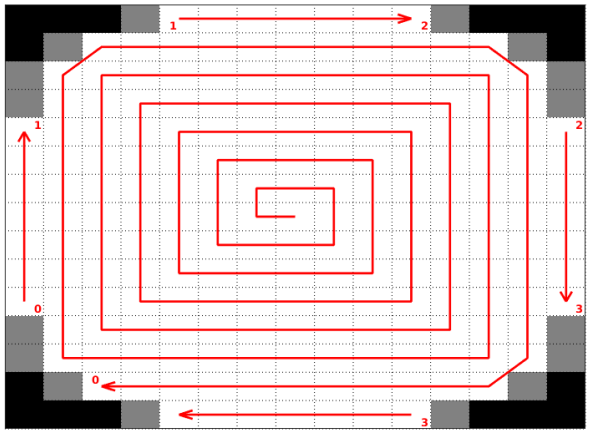



Figure 3.15: View of the matrix of SAIs with the pattern of progression used to create the subjective test videos. Black and dark corner images are ignored for comfort to avoid flickering.

does not impact the noise level significantly, since the hot pixel noise is very different from AWGN. The noise level is again slightly decreased after colour correction, but overall the order of magnitude of the noise level remains unchanged for all these steps. In some cases the noise is even amplified after the colour correction, which further justifies applying denoising last, e.g. *cchart*, *chezed*. A clear reduction of the noise level is observed for all approaches after applying the LFBM5D filter. Overall, our full pipeline provides the smallest noise level compared to applying denoising on the demultiplexing of Dansereau et al. [5] or on our proposed demultiplexing approach. A visual comparison before and after denoising is shown in Figure 3.13.

### 3.8.3 Subjective Evaluation

We evaluated the pipeline using a subjective experiment. For this we crafted videos showing all SAIs, starting from the centre view and following an expanding snail-like pattern going clockwise toward the external views (see Figure 3.15).

This pattern was chosen instead of a more traditional snake-like pattern going from line to line because it highlighted our modifications of the external views more clearly, and offered smoother transitions. To stay consistent across all methods, and to reduce discomfort, we decided to ignore the four black and four dark views in each corner. Only our recolouring

step fixes the dark views and keeping them in the videos causes unnecessary flickering for the other methods. The videos were created with 25 fps for comfort and were therefore approximately seven seconds long. Using the datasets described in Section 3.8, this resulted in 119 videos so the full session lasted approximately 30 minutes, including time for explanations, setup, a short training and comments at the end.

We collected data from 22 voluntary participants (14 men, 8 women) who were tested for visual acuity and colour blindness, and rated these videos in a traditional side-by-side pairwise comparison experiment. We used Psychtoolbox for Matlab in order to ensure the videos were properly synchronised. To reduce bias, we asked the participants to rate the videos based on their personal appreciation, instead of guiding them toward looking for specific artefacts or particular sets of colours. The only emphasis was put on image quality consistency along the videos. The participants were then guided through a short training session to ensure they understood the task at hand, and the controls to perform it. The experiment took place in a dark room as recommended by ITU [129]. The screen was colour-calibrated beforehand.

The responses were processed using a freely accessible tool performing Thurstonian Case V scaling for pair-wise comparison experiments developed by Perez-Ortiz et al. [130]. After scaling, just-objectionable-difference scores (JOD), as described by Perez-Ortiz et al., are obtained for each case. A difference of 1 JOD means that one option is selected over another with 75% probability (the mid-point between random guess and certainty). The relationship between the preference probabilities and the JOD follows the Gaussian cumulative distribution function, and the exact JOD values are found through a maximum likelihood estimation as explained in the work of Perez-Ortiz et al. The outcome is summarised in Table 3.3 and Figure 3.16.

The results show that our demultiplexing is preferred by the subjects more than the one by Dansereau. However, occasionally, some participants commented that they preferred the over-saturated colours obtained with Dansereau et al.'s [5] method more than ours. The results indicate that our hot pixel removal tool has a positive effect of similar magnitude when ap-

Table 3.3: Subjective experiment results: just-objectionable-difference (JOD). First column is 0 we use it as reference for comparison. Negative values indicate the reference (in this case $Da$) was preferred over the method, while positive values indicate the method was preferred over the reference. For explanation and settings details refer to Section 3.8.3 and Table 3.1.

| | $Da$ | $De$ | $DeH$ | $Re$ | $DaN$ | $DeN$ | $ReN$ |
|---|---|---|---|---|---|---|---|
| anky | 0 | 0.64 | 0.61 | 2.05 | -0.17 | 0.8 | **3.07** |
| cchart | 0 | 0.09 | 1.26 | 1.95 | -0.1 | 0.38 | **2.44** |
| desk | 0 | -0.01 | -0.2 | 0.04 | 0.14 | 0.28 | **1.11** |
| friends | 0 | 0.8 | 0.7 | **2.21** | 0.92 | 1.31 | 2.11 |
| magnets | 0 | 0.67 | 1.25 | 2.69 | 0.84 | 0.94 | **3.37** |
| bee_1 | 0 | 1.35 | 2.92 | 3.99 | 0.7 | 2.7 | **4.89** |
| bee_2 | 0 | 7.95 | 8.1 | 10.03 | 0.33 | 8.75 | **10.14** |
| chezed | 0 | -0.03 | -0.16 | 1.12 | 0.41 | 0.05 | **1.26** |
| duck | 0 | 0.01 | 0.22 | 1.14 | **1.16** | 0.1 | 0.83 |
| fruits | 0 | 0.06 | -0.63 | 1.05 | 0 | -0.72 | **1.69** |
| rose | 0 | 0.55 | 0.49 | 1.5 | 0.71 | 0.25 | **1.63** |
| la_guin | 0 | 0.48 | 1.13 | 2.31 | 0.22 | 1.53 | **3.51** |
| chicken | 0 | 0.14 | 1.37 | 2.59 | 1.2 | 0.96 | **2.89** |
| odette | 0 | -0.13 | 0.23 | 0.77 | **1.58** | 0.13 | 1.36 |
| raoul | 0 | 0.94 | 2.82 | 5.26 | 1.19 | 0.84 | **6.45** |
| rodo | 0 | 0.95 | 1.24 | **2.54** | -0.44 | 1.54 | 1.98 |
| ukulele | 0 | -1.15 | -0.5 | 0.59 | 0.84 | 0.44 | **1.15** |
| Overall | 0 | 0.32 | 0.64 | 1.72 | 0.55 | 0.72 | **2.06** |

plied to our demultiplexing. The colour correction step has the biggest effect on the pleasing factor, against all other settings, but even more significantly when associated with the previous steps of our pipeline. Finally, our final denoising step, in all scenarios, shows a level of improvement comparable to that of our demultiplexing and hot pixel removal tool. Overall, we can conclude that SAIs processed using our pipeline are significantly more appealing than when processed with the toolbox of Dansereau et al.

The significance of the results were also analysed by the statistical significance analysis proposed by Perez-Ortiz et al. and reported in Figure 3.16. In this figure, the face values indicate the JOD difference, $JOD_i - JOD_j$, between the $i^{th}$ row and $j^{th}$ column, where positive values indicate that the settings in the row are better than that in the column and negative values indicate the opposite. Black boxes indicate this difference is statis-

| | Da | De | DeH | Re | DaN | DeN | ReN |
|---|---|---|---|---|---|---|---|
| Da | 0 | -0.31 | -0.64 | -1.72 | -0.55 | -0.72 | -2.06 |
| De | **0.31** | 0 | -0.33 | -1.41 | -0.23 | -0.4 | -1.74 |
| DeH | **0.64** | **0.33** | 0 | -1.08 | 0.1 | -0.07 | -1.41 |
| Re | **1.72** | **1.41** | **1.08** | 0 | **1.17** | **1** | -0.34 |
| DaN | **0.55** | 0.23 | -0.1 | -1.17 | 0 | -0.17 | -1.51 |
| DeN | **0.72** | **0.4** | 0.07 | -1 | 0.17 | 0 | -1.34 |
| ReN | **2.06** | **1.74** | **1.41** | **0.34** | **1.51** | **1.34** | 0 |

Figure 3.16: Overall JOD score differences for all contents and subjects, where the face value indicates $JOD_i - JOD_j$, between the $i^{th}$ row and $j^{th}$ column. Positive values indicate the settings of the row are better than that of the column. Black boxes specify that this difference is statistically significant. Refer to Section 3.8.3 for analysis.

tically significant. The results show that overall, all of the proposed steps bring a statistically significant difference compared to the previous step. We can easily see that the whole pipeline (i.e., $ReN$) is superior to all cases, and recolouring is also found to be significantly better than the $DaN$ and $DeN$ cases, which shows that the effect of recolouring is critical for human perception.

### 3.8.4 Aesthetic Appeal

As an additional way to compare our results to the previous state of the art, we use a recent neural metric by Talebi et al. [131] that focuses on the aesthetic aspect of images called NIMA. A summary of this analysis can be found in Table 3.4.

NIMA simulates an estimation of a group of people's ratings for aesthetic appeal based on its pleasing factor, and thus gives an average score as well as standard deviation for each image. We obtain our results by testing each individual SAI, and average the results to get a unique score for each light field. NIMA can also be used as a metric to measure noise level, but

Table 3.4: NIMA results. For better reading, indicated in bold black are the best scores, and in italic blue the worst ones. The values represent for each content the average of individual views' scores. For settings details refer to Section 3.8.3 and Table 3.1.

| | $Da$ | $De$ | $DeH$ | $Re$ | $DaN$ | $DeN$ | $ReN$ |
|---|---|---|---|---|---|---|---|
| anky | 4.67 | *4.57* | *4.57* | 4.81 | 5.49 | 5.47 | **5.56** |
| cchart | *5* | 5.05 | 5.05 | 5.07 | 5.23 | 5.3 | **5.32** |
| desk | *4.7* | 4.78 | 4.78 | 4.83 | 4.99 | 5.05 | **5.16** |
| friends | *5.15* | 5.4 | 5.39 | 5.52 | 5.25 | 5.47 | **5.61** |
| magnets | 4.22 | *4.12* | *4.12* | 4.27 | 4.96 | 4.85 | **4.97** |
| bee_1 | *4.26* | 4.35 | 4.35 | 4.46 | 4.53 | 4.77 | **5.05** |
| bee_2 | 4.5 | *4.36* | *4.36* | 4.46 | 4.47 | 4.46 | **4.6** |
| chezed | *5.27* | 5.33 | 5.29 | 5.32 | 5.34 | 5.4 | **5.47** |
| duck | *4.84* | 4.86 | 4.85 | 5 | 4.96 | 4.98 | **5.12** |
| fruits | **4.66** | 4.52 | 4.52 | 4.52 | 4.63 | *4.46* | 4.47 |
| rose | **4.93** | 4.88 | 4.88 | 4.84 | 4.75 | 4.67 | *4.66* |
| la_guin | 4.66 | *4.3* | 4.32 | 4.44 | 4.92 | 4.95 | **5.06** |
| chicken | *3.98* | 4.03 | 4.02 | 4.18 | 4.06 | 4.73 | **4.95** |
| odette | 4.89 | *4.78* | 4.79 | 4.87 | **4.97** | 4.84 | 4.95 |
| raoul | 4.24 | 4.16 | 4.16 | 4.31 | *4.12* | 4.19 | **4.7** |
| rodo | 4.38 | 4.38 | 4.37 | *4.36* | **4.39** | 4.38 | 4.37 |
| ukulele | 4.55 | *4.5* | *4.5* | 4.58 | 5.36 | 5.19 | **5.33** |
| Average | 4.64 | 4.61 | *4.61* | 4.7 | 4.85 | 4.89 | **5.02** |

since we are interested in the pleasing factor and have more dedicated metrics for noise analysis, we decided to use it by resizing the images instead. As suggested by the authors, each SAI is resized from $625 \times 434$ to $224 \times 224$ before being evaluated by the pre-trained network, since this allows for the most accurate results based on aesthetic quality.

From Table 3.4, we can see that, with few exceptions, the results obtained using our full pipeline garner better scores compared to those processed by the toolbox of Dansereau et al. [5]. On average, both the recolouring and denoising step improve the image quality, except in the case of the *fruits* and *rose* light fields in which Dansereau et al.'s method performs better. Images obtained with Dansereau et al.'s method have brighter, more saturated colours than those generated using our approach and the NIMA network can associate these unnatural colours with better aesthetic value. This is consistent with comments made by some participants of the subjective experiment described in Section 3.8.3.

### 3.8.5 Computation time

We report here the average computation times for each part of the pipeline. Most of the steps were implemented in Matlab, and the denoising was implemented in C++. Our demultiplexing step takes ~2'05" per light field, whereas in comparison the demultiplexing of Dansereau et al. takes ~1'10". The difference is essentially explained by the White Image-guided interpolation. The HPR step runs in ~1'40". Correspondences between neighbour views and with centre view (2 sets per view to recolour) are computed in ~5'45". The recolouring step runs in ~234' (~60" per SAI) and finally the denoising step takes ~50'.

Possible optimisation includes parallelisation of the colour correction step, as several rows could be processed at the same time, once the centre column images are available. GPU implementation would also speed up the process of the propagation step, or the denoising. Finally, our implementation of the recolouring uses all the available correspondences, when a fraction could be selected to reduce the computation time, albeit with reduced quality. Finally, in this work we have proposed using CIELAB space colour values when estimating the colour transfer function to ensure the best results. Reducing the colour space representation from three channels to two could also provide significant computational speed up and would be an interesting avenue for future investigation.

## 3.9 Applications

### 3.9.1 Rendering

One of the first light field applications was the ability to synthesise new images corresponding to novel viewpoints in real time, without requiring any 3D model of the scene, as described by Levoy et al. [10]. For each pixel in the novel image, the intersection of the corresponding light ray and the two light field planes is computed. The intersection with the camera planes allows the closest available SAIs to be found, while the closest pixel positions are computed from the intersection with the image plane. The final value of a pixel in the novel image is then computed by interpolating between the nearest SAIs and the nearest pixels.

Figure 3.17: Novel viewpoints rendered from the *cchart* light field, moving from left to right. Top: Dansereau et al. [5] ($Da$). Bottom: ours ($ReN$). Colour inconsistencies inside and across viewpoints are highlighted in red.

In this experiment, we rendered novel views corresponding to a camera close to the object of interest and moving horizontally from left to right. We show a few rendered images for the *cchart* and *bee_2* light fields in Figures 3.17 and 3.18 respectively. On the top row, results obtained for a light field decoded with the toolbox of Dansereau et al.[5] ($Da$) are displayed, and on the bottom row results obtained with our full pipeline ($ReN$). As rendered images are created from multiple source SAIs, clear colour inconsistencies appear in images rendered from Dansereau, but also in between the different novel viewpoints. In addition, images rendered from our pipeline are less affected by the dark SAIs in the corners of the light field.

### 3.9.2 Compression

Due to the large amount of information contained in light fields, their compression is essential for a large scale adoption of this image format.

Figure 3.18: Novel viewpoints rendered from the *bee_2* light field, moving from left to right. Top: Dansereau et al. [5] ($Da$). Bottom: our full pipeline ($ReN$). Colour inconsistencies inside and across viewpoints are highlighted in red.

However, aforementioned artefacts in existing plenoptic data are likely to reduce the efficiency of traditional compression methods. In order to evaluate the impact of our quality enhancement tools on the compression performance, we have used a common light field compression method presented by Liu et al. [132]. This method forms a pseudo video sequence from the light field's SAIs and encodes the sequence using the HEVC video coding standard, therefore taking advantage of redundancies between SAIs.

For this experiment, we have encoded three different versions of each light field corresponding to $Da$, $De$ and $Re$ in Table 3.1 (i.e. demultiplexing of Dansereau et al. [5], our demultiplexing only, and our demultiplexing followed by hot pixel removal and colour consistency correction). Each version was encoded several times with different bitrates by varying the QP parameters in HEVC over the values $\{12, 16, 20, 24, 28, 32, 36\}$. In order to evaluate the quality of the decoded light field, we compute the peak signal to noise ratio (PSNR) using as a reference, the uncompressed light field of the corresponding version. The experiment was performed for 12 light fields including 4 from the EPFL dataset, 4 from the INRIA dataset

Table 3.5: Bitrate savings obtained for light fields extracted with our demultiplexing ($De$) and with our hot pixel removal and colour correction ($Re$). The gains are computed with the Bjontegaard metric [133] with respect to light fields extracted using the method of Dansereau et al. [5] ($Da$). These results assume that similar PSNR for each version ($Da$, $De$, $Re$) correspond to similar perceived quality.

| $Source$ | $Image$ | $De$ | $Re$ |
|---|---|---|---|
| EPFL | bikes | -0.9% | -29.1% |
| | fountain&vincent_2 | 8.9% | -33.7% |
| | stone_pillars_outside | -19.8% | -59.2% |
| | vespa | 10.8% | -50.1% |
| INRIA | bee_2 | -67.7% | -92.2% |
| | bumblebee | -36.1% | -78.1% |
| | duck | -52.1% | -80.3% |
| | fruits | -62.3% | -81.1% |
| V-SENSE | cherry_tree | -35.8% | -55.4% |
| | chicken | -83.1% | -98.7% |
| | rodo | -51.2% | -72.2% |
| | wine_bottles | -67.3% | -93.4% |
| **Average** | | **-38%** | **-68.6%** |

and 4 from our captures (V-SENSE).

Note that the PSNR is computed from a different uncompressed reference for each version. However, our experiments in Section 3.8 have shown that our modified demultiplexing as well as our additional hot pixel removal and colour consistency correction steps improve the subjective quality in the uncompressed case. Here, we assume that the relative perceived quality of the three version $Da$, $De$ and $Re$ are unchanged when they are altered with similar compression losses. Therefore, we consider that for the same PSNR scores, the quality of the compressed light fields $De$ and $Re$ will not be perceived as worse than that of $Da$. Furthermore, the results in Table 3.5 show that, on average, the light fields in $De$ and $Re$ require respectively $38\%$ and $68.6\%$ less bitrate to be encoded with a similar PSNR as $Da$. This clearly demonstrates that the enhanced quality resulting from both our demultiplexing and post processing steps also has a very beneficial impact on the light field compression.

### 3.9.3 Super-Resolution

Light fields captured by lenslet cameras have a poor spatial resolution due to the multiplexing of both spatial and angular information on a single sensor. Spatial super-resolution of light fields captured with a lenslet camera is thus a common application.

In this experiment, we used the extension of the LFBM5D denoising filter to spatial super-resolution presented by Alain et al. [134]. This method uses the sparse coding of the LFBM5D filter as a prior to solve the ill-posedness of super-resolution. A two-step iterative algorithm alternating between a LBM5D filtering step and a back-projection step is used to obtain the super-resolved light field.

We show results for a single SAI of the *raoul* light field in Figure 3.19. The super-resolution result (right) is compared to a simple bicubic upsampling (left). Results obtained with the toolbox of Dansereau et al. [5] ($Da$) are displayed on the top row, with results for our full pipeline ($ReN$) on the bottom row. The benefit of our pipeline is clearly visible, especially in terms of hot pixels and noise removal. This is due to a general side effect of super-resolution which amplifies the high frequency corresponding to noise. This is common to all super-resolution methods, not only the one used here.

### 3.9.4 Light Field Editing

Light field editing is another important application in light field imaging, with works by Jarabo et al. [15] or Zhang et al. [135]. To determine whether our proposed pipeline provides any advantages for light field editing applications, we applied the recent editing technique of Frigo et al. [136] to both our processed light fields and those processed with Dansereau et al.'s method [5]. The technique proposed by Frigo et al. [136] allows the user to edit the centre SAI of the light field, either via image recolouring or inpainting, and propagates the edits to the remaining views using a structure tensor driven diffusion on the EPIs. Some light field editing results can be seen in Figures 3.20 and 3.21.

Due to the strong colour differences between the centre SAI and the external views of light fields obtained using Dansereau et al.'s method (*Da*), the

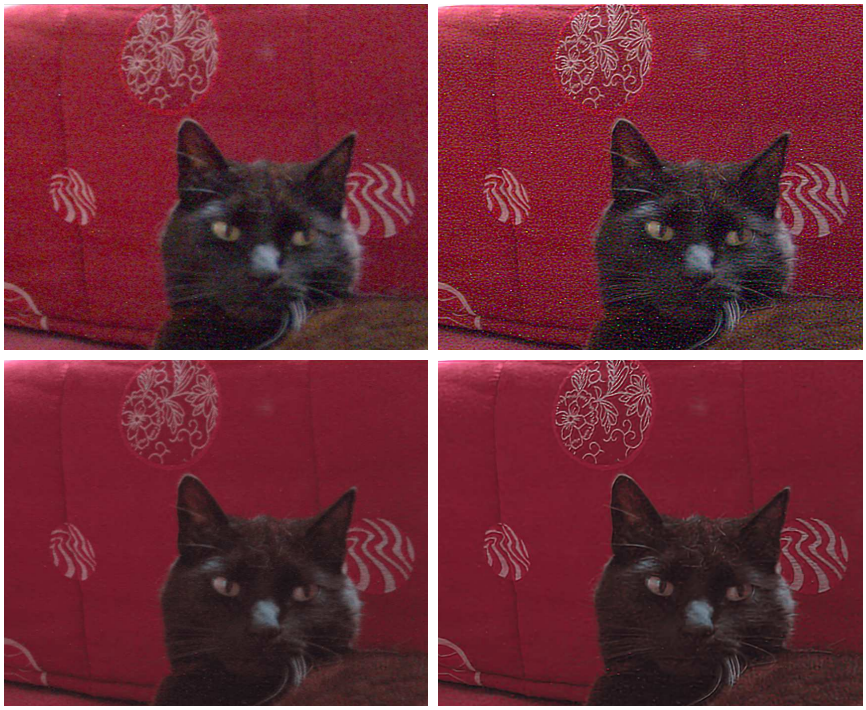


Figure 3.19: Spatial super-resolution (right) of the *raoul* light field compared to a simple bicubic upsampling (left). Top: Dansereau et al. [5] ($Da$). Bottom: our full pipeline ($ReN$).

tensor-driven diffusion becomes inaccurate at the edges of the light field, causing unwanted warping of the SAIs (Figures 3.20 and 3.21, column 1). The strong colour differences between SAIs also means that when colours from the centre SAI are propagated to other SAIs, they do not blend seamlessly with the rest of the image, creating strong colour inconsistencies (Figures 3.20 and 3.21, column 1, see inpainting results). Interestingly, we also found that when editing light fields generated using our full pipeline, including denoising (*ReN*), unwanted warping artefacts are also created (Figures 3.20 and 3.21, column 3). As with any denoising algorithm, small image details can also be removed with noise, some of which are needed by the tensor diffusion step in the edit propagation software proposed by Frigo et al. Removing these details creates inaccuracies and causes artefacts. On the other hand, edit propagation results applied to our pipeline before denoising (*Re*) are the best (Figures 3.20 and 3.21, column 2). The consistent colours across these light fields ensure that the edits are propagated correctly, and that no inconsistent colours can be seen in the edited SAIs, even towards the outside of the light field. This indicates that if using

a similar editing approach, edit propagation should be applied after our recolouring step, with denoising applied as a final step.

### 3.9.5 Depth / disparity estimation

We evaluate here the performance of the proposed pipeline on depth or disparity estimation, which is one of the flagship applications for light fields. For that purpose we use 4 different methods [6][7][8][9] applied after every step of the pipeline. For all methods we used the code provided by the authors. The first method estimated the depth by simply computing the slopes of the EPIs based on the light field gradient [6]. Note that the code provided by the authors implements the first step described in the paper and only outputs a sparse estimation. The second method was designed to be robust to occlusions by analysing the statistics of angular patches of the light field together with refocus cues [7]. The third method uses the spinning parallelogram operator to estimate the slopes of the EPIs and provide a robust depth estimate [8]. Finally, the fourth method adapted optical flow techniques to estimate the disparity on row or columns of the light field [9].

Figure 3.22 shows the results for the four methods on the *bee_2* light field. Results for 7 additional light fields are available in the appendix. For each method, the depth or disparity was estimated for the centre SAI of the light field decoded with the toolbox of Dansereau et al. [6] without (*Da*) and with denoising (*DaN*), our demultiplexing (*De*), and our full pipeline without (*Re*) and with denoising (*ReN*). Note that all results were colour coded so that close objects appear white, while far objects appear black.

Since no ground truth is available for the depth or disparity maps, no objective evaluation could be conducted. For each method, slight variations can be observed between the depth or disparity maps corresponding to the different steps, but no step seems to clearly deter or improve the performances. Note that this is also true after the denoising step, even though denoising is sometimes not recommended before such applications. While in general denoising may smooth images, the LFBM5D algorithm chosen for this work can preserve edges, which are useful features for most depth or disparity estimation methods. Thus the proposed pipeline does not seem to strongly impact the performances of depth or

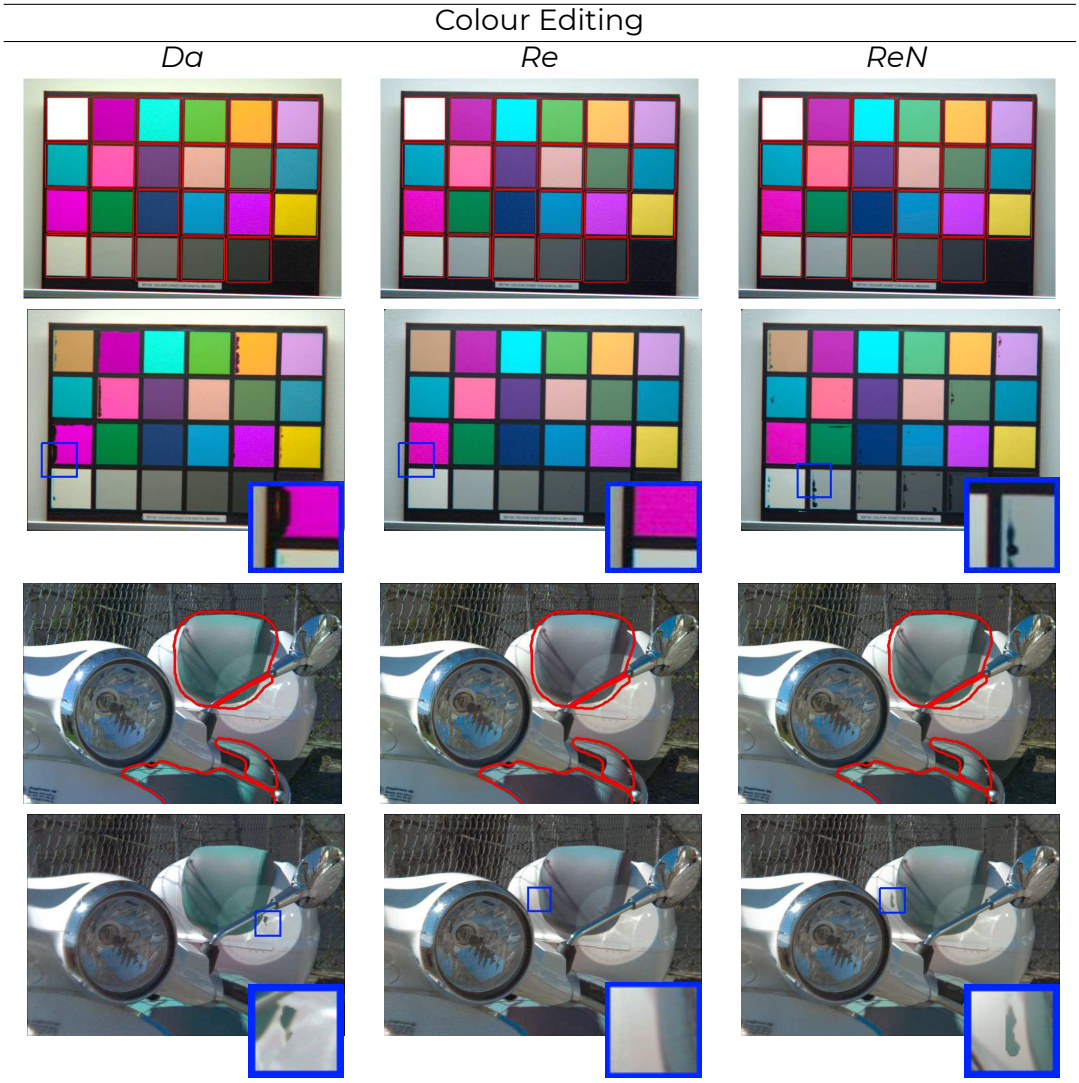



Figure 3.20: Light field colour editing results using the edit propagation method of Frigo et al. [136]. For each light field, the top row shows the user edits made to the centre SAI of the light field, with red lines indicating the mask used during the propagation process. The second row shows a sample SAI from the light field after the edit propagation.

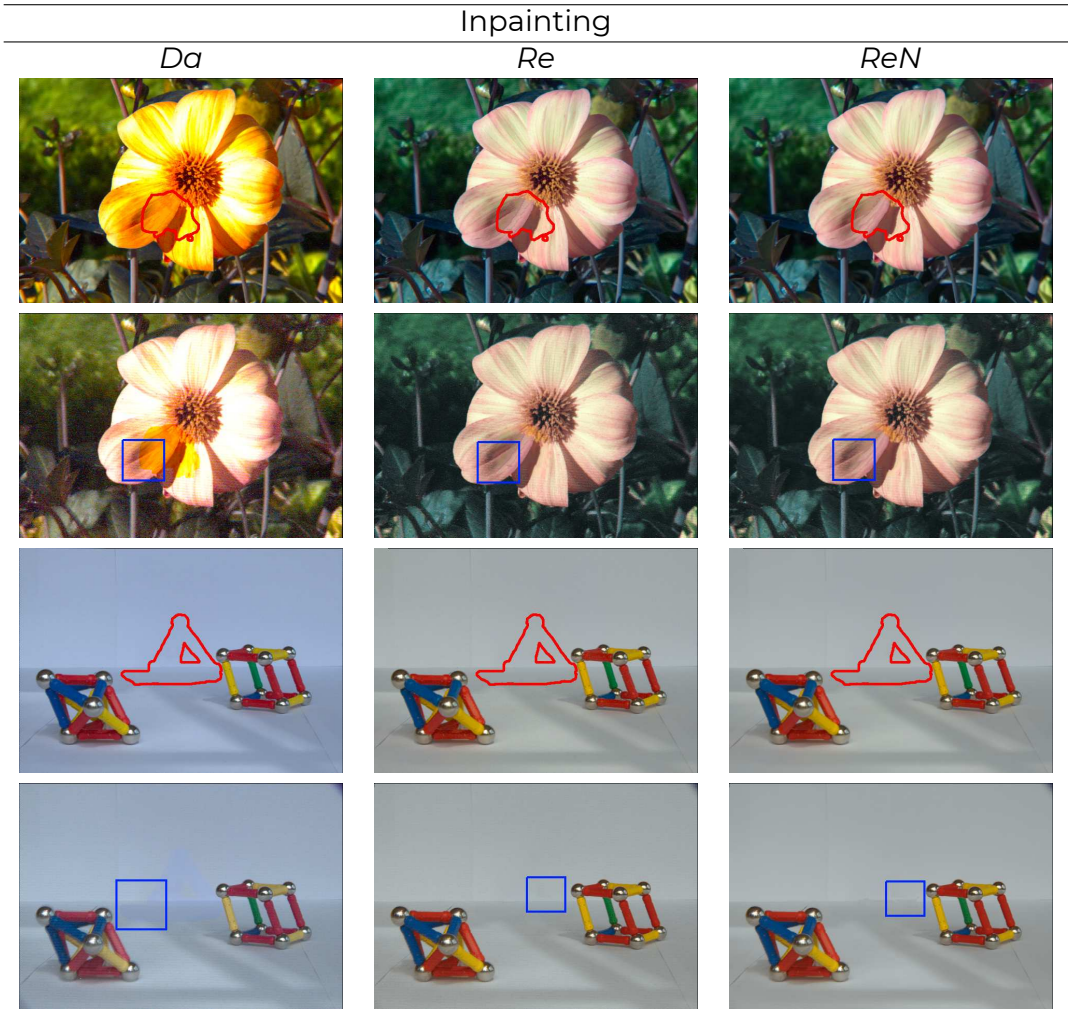



Figure 3.21: Light field inpainting results using the edit propagation method of Frigo et al. [136]. As previously, for each light field, the top row shows the user edits made to the centre SAI of the light field, with red lines indicating the mask used during the propagation process. The second row shows a sample SAI from the light field after the edit propagation.

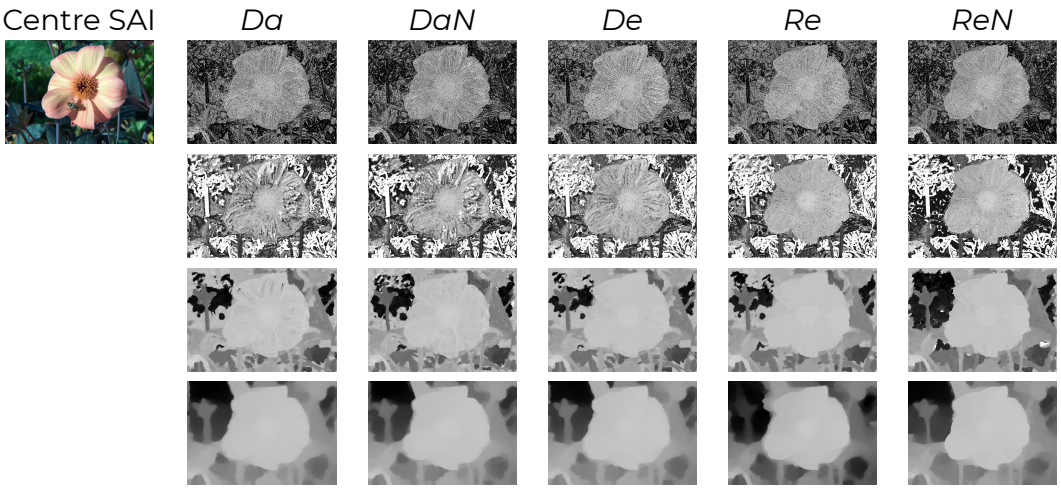



Figure 3.22: Depth maps estimated for different steps of the pipeline on *bee_2* using, from top to bottom: [6], [7], [8], and [9].

disparity map estimation.

## 3.10 Conclusion

We presented in this chapter a high quality light field extraction pipeline aimed at reducing or removing the various artefacts, colour inconsistencies and noise that are prevalent in the typical output from plenoptic cameras. We provide and analyse several tools that can be used either on their own or in conjunction with each other for increased effect, and we show that each of the steps is necessary to ensure the best possible image quality. We also highlight the importance of the order in which each step is performed within the pipeline. We have proven, using a number of metrics, as well as a subjective experiment, that our results outclass those obtained from the previous state of the art tools, and finally make the entirety of the sub-aperture views usable by the user. We note that both the recolouring and denoising steps in our pipeline can be applied to light fields captured with camera arrays or gantries, and are not limited to plenoptic light fields. Finally we demonstrate that using higher quality light fields enhances the quality of the results for a number of classic light field applications, and therefore expect that this improvement will allow the research community to be keener to use these cameras and data for

their work.

# Chapter 4

# Light Field Soft Colour Segmentation

In this chapter we detail our second contribution, which looks at performing soft colour segmentation on the enhanced light field data we obtain using the methods detailed in the previous chapter. We show that this segmentation benefits from additional light field images, and detail another contribution, using depth cues to perform object segmentation. Finally we take look at edited results and show that our contributions help improving the output quality. Finally we put this in perspective and discuss how the methods could be improved.

## 4.1 Introduction

The field of research in light fields is vast and covers a wide array of applications such as rendering, depth estimation, or super-resolution, novel view synthesis, compression, and many more. One field in particular seems to be less attractive to researchers: image editing, which we theorise could take advantage of the higher dimensionality of light field data. In this chapter, we investigate the possibility of applying colour decomposition algorithms on light fields and detail the advantages and drawbacks of such methods.

As far as we are aware, this is the first work looking at using these methods on light field images. Additionally we propose taking advantage of light fields to counteract one of the drawbacks of decomposition methods. Since the output of such algorithms is generally a number of colour layers based on a pre-computed palette, these layers can contain object and background information that have no semantic relation to each other. As a result any editing done on a single layer would affect all these objects, perhaps at the risk of causing unwanted artefacts, e.g. applying unnatural colours to human skin. In this work we present an automatic depth-based object-aware layer separation method to allow for easier colour editing.

## 4.2 Related work

### 4.2.1 Soft colour segmentation

Soft colour segmentation is a method of image decomposition which consists of separating the image into several semi-transparent layers containing pixel information close to a colour from a pre-computed palette. Initial works by Aksoy et al. used colour unmixing to satisfy minimisation functions, in which the colour palettes were computed by probability distributions obtained through pixel voting [16]. Tan et al. obtain a colour palette by simplifying a RGB convex hull of all observable colours [137]. The simplification can be adjusted to obtain a different number of colours in the palette.

Aksoy et al. further improved upon their previous techniques by implementing a more efficient voting scheme [17]. In order to obtain more consistent colour layers, Tan et al. use spatial coherence by extending their palette extraction method through a RGBXY convex hull [138]. A new technique by Koyama et al. decomposes images based on editing software blending modes [139]. The resulting layers may contain colour that do not appear in the original image and only using the proper blending modes for reconstruction to ensure a stable result.

One of the drawbacks of Tan et al.'s method [138] is that by simplifying a RGB convex hull, it is quite possible to obtain palette colours that do not appear in the image. Feeling that obtaining a palette that would not be representative enough of the image can make editing work less intuitive for the user, Wang et al. use a similar method where a polyhedron is placed around the image colours in 3D space [140]. However they do not necessarily compress it to the convex hull and the palette they extract through an optimisation problem ends up being more accurate. Jeong et al. first sample pure colours, then build a hierarchical model by splitting each layer, and all the possible colours within it, into two layers where the colour variance is much smaller and the dominant colours are as distinct from each other as possible. [141]

### 4.2.2 Object Segmentation for Light Fields

Object segmentation on light field images has had a variety of techniques proposed. Mihara et al. propose a graph-cut method that works on sub-aperture views to find object edges and segments objects by enforcing a global consistency [142]. Hog et al. developed a method to exploit the redundancy of light fields to reduce the graph size of Markov random fields by using a ray bundle structure [143]. Their method is interactive and ensures stability and consistency across all light field views. Zhu et al. propose a super-pixel segmentation method which uses ray-tracing in the light field volume and accounts for the disparity between each super-pixel to provide a refocus-invariant segmentation [144]. A more recent method proposed by Khan et al. improves upon the previous one by using a clustering step to enforce better consistency across the light field instead of simply propagating the results from the central view [145].

### 4.2.3 Light Field Editing

Jarabo et al. provided a comprehensive overview of different techniques used for light field editing, which included colour editing, inpainting, adding objects at various depth layers or drawing on partially occluded surfaces [15]. Le Pendu et al. propose a novel method for inpainting using low rank matrix completion which takes advantage of the redundancy of light field views [124]. Frigo et al. developed a method using epipolar planes to propagate edits, colour or inpainting, to the entire light field in a consistent manner [136]. Zhang et al. created a method allowing object manipulation such as resizing or moving through depth planes [146]. They first decompose the central image in different depth layers, allowing the user to edit any of them. The patch-based method then reconstructs the image by transforming all possibly affected layers. Finally these edits are propagated from the centre view to the rest of the light field views. In this work we wish to provide another level of editing for artists and other light field users.

## 4.3 Soft colour segmentation

We base our investigations on our implementation of the more recent method by Aksoy et al. [17]. This decision was motivated by the output of the method, providing layers of colours present in the image, which allows for more intuitive editing. As the method was developed for single images, we require a new strategy to apply it on light field data in the sub-aperture image representation, in order to enforce consistency in both the colour palette and the segmentation between all the views. In this section we describe some of the methods we used in achieving this goal.

### 4.3.1 Naive approach

For this initial approach, we apply the entire soft colour segmentation method to each individual sub-aperture view. That means a colour palette is computed for each image and the layer separation is done based on this palette. Because of slight variations in colour distribution caused by the disparity, the computed palette is not consistent across all views. This is

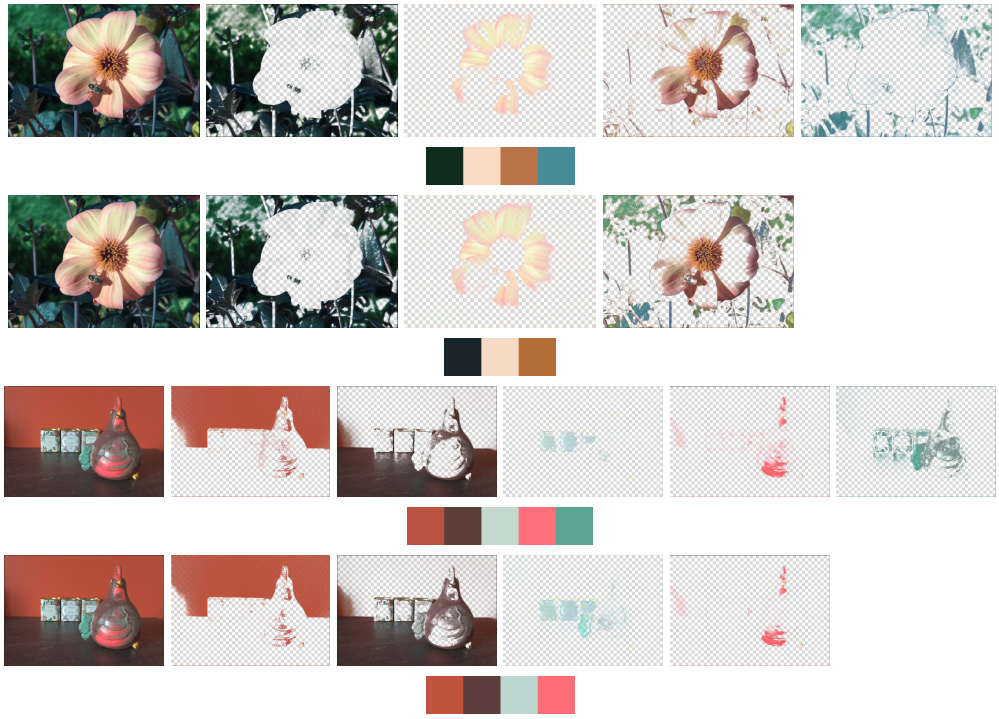


Figure 4.1: Results of our naive approach to perform soft colour segmentation on two consecutive views of images *bee_2* and *chicken*, with associated colour palettes. The original view is on the left. In both cases, as with most images we studied, the number of layers can be different across views. In the top row of image *bee_2* the last two layers end up mostly in a single layer in the decomposition of the bottom row, and in the top row of *chicken* the additional layer is composed mostly of pixels that end up in layers 2 and 3 in the decomposition on the bottom row. This results in major inconsistencies in these layers.

visible in Figure 4.1, where we show the layer decomposition of two consecutive views in a light field row. In many of the examples we worked on, the palette size varies by one or even two colours, leading to layers inconsistent in number and representation. Any kind of editing done with layers like these would result in erroneous results with flickering between views and artefacts clearly visible no matter which method is used to represent or view the light field.

### 4.3.2 Global approach

After seeing the output of our first method, we investigated ways of ensuring global consistency, especially regarding the computation of the colour palette. To this end, we first construct a mosaic image containing all the

Figure 4.2: Results of our global approach to perform soft colour segmentation on the same consecutive views as in Figure 4.1. Compared to the naive approach here the number of layers is equal, and the colour distribution within layers is more consistent.

views from the light field, and perform an initial computation of a single global colour model, before applying the soft colour segmentation to each individual view using that single colour palette. The reasoning behind this was to ensure all colours from the light field would be represented in the colour palette, including some that might appear only in specific views due to occlusions. This method, as we can see in Figure 4.2, produces more spatially consistent results with less variation between views, although some minor flickering can still be detected upon closer inspection (zoom in). The colour distribution in each layer is also generally more consistent and useful in comparison with the results of the naive method where we obtain a different number of layers. When using this global method we enforce consistency in both colour palette size and within the composition of each layer.

Using a global colour model additionally ensures all the views have the same number of layers and allows for situations where some objects could suffer from occlusion in some views but not others. In the case where the object is occluded, the layer would simply appear nearly empty. This is preferable to having the occluded objects appear in unrelated colour layers because of their low representation ratio.

### 4.3.3 Epipolar plane images

We additionally attempted to perform soft colour segmentation on epipolar plane images (EPI) instead of the sub-aperture views. We compute the colour palette using information from the whole light field rather than from a select view. Similar to the global method, the results are much more consistent globally, since the EPIs contain exactly the same colour information as the related views, even though the distribution is different. As this method offers no advantage while adding an extra computational step to generate the EPIs, we decide to use the method described in section 4.3.2 as our base for the rest of this chapter.

## 4.4 Object-based layer separation

In this section we present a method to separate objects in layers based on their depth. When analysing the results of traditional soft colour segmentation, it appears that many layers contain information from different objects which do not necessarily have a semantic relation. Editing the entire layer will affect all these objects and may cause some undesirable artefacts. Our intuition is to separate some of these layers into semantically relevant ones, which should make some editing tasks easier to perform.

As light fields give us additional depth information, we chose to use it to separate objects that may appear in the same colour layers. We use the Spinning Parallelogram Operator method of Zhang et al. [8] to obtain a depth map of the light field. Using this information, we compute a histogram of the depth values of the image, as in Figure 4.3. Once we obtain the histogram, we assume each peak represents a specific object, or at

the least a depth layer containing mostly information from a single object. Values in and around the minima typically mark the separation. However to get the most precise split we use a gradient-based method described in the next paragraph. For instance in Figure 4.3, it would appear there are three potential separate objects based on the histogram. Visual inspection, however, only shows two objects, while the values near zero belong to the background.

To rectify this as well as cases where there exists some overlap in depth between separate objects, we use a gradient-based metric which looks at continuity between the approximate centroids of the objects and the rest of the pixels. Starting from these centroids, we go outward and measure the difference in depth values until we detect sudden changes larger than a user-defined threshold typically chosen based on the range of depth values. Experimentation shows that setting this value as half the spread of depth values for the current object gives satisfying results. If that threshold is exceeded we determine the pixels belong to another object. Some visual results of layer splitting based on this can be observed in Figure 4.4.

Even though the method splits the layer into separate, semantically coherent layers, some artefacts may occur. For instance in the image $greek$ the layer containing the right statue still contains some background information. This is due to the depth estimation method incorrectly handling this boundary. Similarly in the image $chicken$ some information from the metal boxes end up in the same layer as the background wall, instead of being in a separate layer, since the depth estimation puts them both on the same level.

## 4.5 Experimental results

In this section we briefly detail the run time of the different parts of our work. We additionally present edited results taking advantage of our contributions. The images in this section have been sampled from the INRIA dataset [124], the HCI dataset [147] and some are our own Lytro images [20]. All of the Lytro data has been decoded, colour corrected and denoised using our methods decribed in chapter 3.

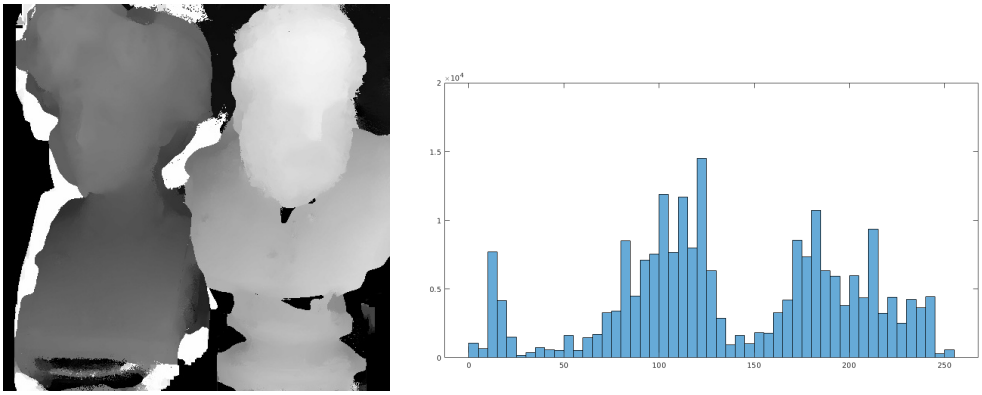



Figure 4.3: Depth map of synthetic image $greek$ (see Figure 4.4). On the right, the histogram of depth values, cleaned to ignore the white values around the left statue and the black values belonging to the background.

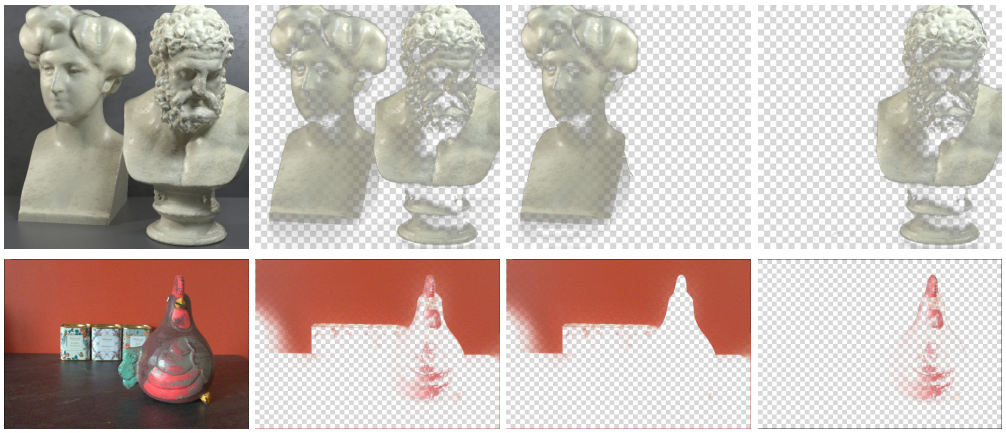


Figure 4.4: Examples of layer splitting using depth information on synthetic image $greek$ and real image $chicken$ The original view is on the left.

### 4.5.1 Computation time

The computation time for a single light field view using the naive approach, i.e. computing the colour palette and doing the segmentation, takes on average $12$ minutes for our C++ implementation of the method, which has not been optimised. This needs to be multiplied by the number of usable views in a light field, which can go up to $209$ for Lytro images.

Computing a colour palette using our global method takes roughly $2.5$ minutes on an image tile of size $5000$x$5000$. While long, this step has to be done only once per light field and the results can be saved for later use. To put this in perspective, computing the individual palettes takes on aver-

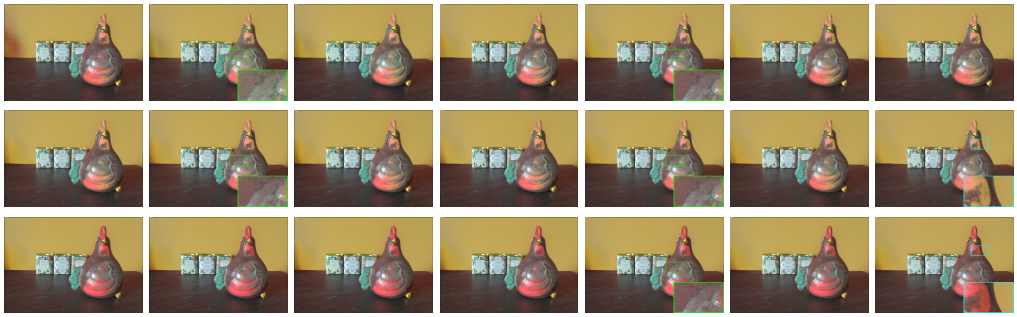


Figure 4.5: Editing results on the main red layers (the first one in Figure4.1 or 4.2) of several sub-aperture images of a light field row; each column represents the same view. On the top row we use layers obtained using our naive approach, and inconsistencies can be observed between the views, on the chicken figurine or the background. In the middle row we use the layers from our global approach, the results are consistent across views but editing the wall colour results in changes affecting the red portions of the figurine. The last row shows editing on the global approach layers, done after separating the first red layer to isolate the wall from the figurine. Here the results are consistent across views, the figurine is unaltered, and we obtain the effect we were aiming for.

age $14$ minutes for a Lytro image ($4.05$ seconds x $209$ views) and produces inconsistent results. However, the more computationally intensive part to perform the soft colour segmentation on each view still takes $12$ minutes on average. An intended future work will look at ways to initialise the segmentation using results from already processed central views and propagate them toward the edges of the light field, in an attempt to reduce the time needed by the optimisation method of the soft segmentation to reach a solution.

Splitting the layers to contain only one object is done through a MATLAB script and is a much faster process, taking on average $2$ seconds per view, regardless of the number of objects.

### 4.5.2 Layer editing

We present results of colour editing on specific colour layers obtained with both the naive and global methods, before and after splitting them, shown in Figure 4.5. Here the edit was to change the background colour to increase the contrast with the foreground object being the centre of the image. Editing using the layers from our naive approach results, as expected, in visible inconsistencies between the views. When using the layers from our global approach, as the main red layer contains pixels from

Figure 4.6: Example of a failure case (real image $guinness$), where the pink reflection and refraction on the glass coming from different objects is difficult to separate from the pen using our method. For readability not all segmented layers are shown, as many objects in the image contain shades of pink.

both the background and the chicken figurine, editing the background without splitting the layer results in unwanted alterations, as it changes the colour of the object as well. However, when the relevant layer is properly segmented to contain only the chicken figurine (or the background), edits are easier to perform and avoid unwanted side-effects. We obtain the intended effect to increase the colour contrast and enhance the focus on the object.

### 4.5.3 Failure cases

Unfortunately there are cases where our method for splitting a layer is not robust enough. One such case is when non-Lambertian objects are present, such as glass, see Figure 4.6. Here the pink layer contains pixels from the pen, some items in the background, and reflection and refraction from all of these on the glass itself. Splitting the layer using our method puts both the pen and the glass in the same layer, and the background objects in their own layer. Editing either of those layers would result in inconsistencies between the background objects and their refraction in the glass. A future work will be to look at better methods to handle these cases.

## 4.6 Conclusion

We presented in this chapter a new method to perform soft colour segmentation on light field data. We have shown that our global approach is well-suited to create high quality colour layers consistent across all views of a light field. We also took advantage of light field data to further separate colour layers based on their content, using depth information, in order to allow for easier editing while reducing the amount of side ef-

fects.

We have also explained that these methods are unfortunately very costly in computing time and future investigations may look at ways to use disparity-adjusted layer decomposition results from neighbouring views as an initialisation step, in order to bring the minimisation algorithm closer to a solution. This should hopefully lead to a significant reduction in computing time when the method is applied over the whole light field. Future work may also include looking at more robust and automatic methods for object-based layer separation and ways to extend this concept to light field videos.

# Chapter 5

# Comparing Traditional Light Field methods with NeRF

In this chapter we detail our third contribution, whereby we compare and detail two classical light field applications, view synthesis and depth estimation, with NeRF, a newcomer in the field, which promises high quality novel views with a minimal setup. We show that indeed NeRF has the advantage when looking at the number and the quality of the results, however we put these in perspective, by detailing the mandatory manipulation needed to use it with certain type of light field data, as well as the much higher computational cost, which could be a deterrent to some.

## 5.1 Introduction

NeRF is a recent deep-learning based method [18] that created a small revolution in how light fields are thought of, constructed, and processed. While most previous light field data is precisely defined, with specific parameters, fixed baseline between views, and a number of constraints, NeRF instead works from using only a relatively small number of images pointing at the same scene from different angles. After some training of the underlying network, it builds a 3D representation of the scene, from which additional information can be extracted, such as novel views, or disparity maps. This is based on the concept of unstructured light fields.

While this new technology seemed almost miraculous when it was first presented, it also seems to suffer, in a way not dissimilar to traditional light field methods, from some drawbacks in how it can be used, and in particular the type of data it can be used with. While traditional light field methods struggle to generalise to data containing a very wide baseline or very high resolution, NeRF on the other hand requires the principal point of the images to be centred. This is not the case with data captured with plenoptic cameras, or digitally synthesised images, and those require some minor additional processing before being used by NeRF.

This chapter aims at looking in more detail and comparing the respective output and failure cases and both traditional light field methods and NeRF when applied to two applications: view synthesis and depth estimation.

## 5.2 Related work

In this section we briefly describe the current state of the art regarding traditional light field view synthesis as well as depth estimation, and relevant papers using NeRF for those same applications.

### 5.2.1 Light Field view synthesis

We first have a brief look at classical computer vision methods. Shi et al. [148], after observing that the sparsity is much greater in the continu-

ous Fourier spectrum than the discrete spectrum, proposed an approach to reconstructing views optimised for sparsity in the continuous Fourier spectrum, to reduce sampling requirements and improve quality. Chen et al. [9] produced consistent disparity maps using the combination of a feature flow method and a spatio-temporal edge-aware filter. Vagharshakyan et al. [149] use the sparse representation of Epipolar Plane Images (EPI) in the shearlet transform domain and take advantage of the straight line characteristic of EPIs for reconstruction. In particular their method handles semi-transparent objects in a scene with a much higher degree of precision.

Kalantari et al. [150] were among the first to use machine learning to mitigate the usual trade-off between spatial and angular resolution of plenoptic cameras. They break down the view synthesis process into disparity and colour estimation components trained simultaneously to obtain high quality reconstruction. Wang et al. [151] present a 4DCNN network combining convolutions on stacked EPIs, and detail-restoration 3DCNNs to effectively synthesise 4D light fields from a sparse selection of views. Yeung et al. [152] use a coarse to fine scheme to extrapolate high-dimensional spatio-angular features in a two-step method first generating intermediate coarse novel views which are later refined using guided residual learning and 4D convolutions. More recently, Chen et al. [153] look at the data collection drawback of other learning-based approaches, and propose a self-supervised framework. They first train their network on natural videos, and use that prior knowledge combined with a cycle consistency constraint to build a bidirectional mapping and generate input-consistent views.

Predating NeRF in concept, a new technique was developed by Zhou et al. [154], called *multi-plane images* (MPI) and generated some interest as a new representation of light fields. MPIs approximate a light field by generating a stack of semi-transparent coloured layers organised at various depth levels, which allows for real-time synthesis of novel views. Early work was constructing MPIs from dense sets of views, but this was soon generalised to sparser sets of real-life images [155, 156, 57].

### 5.2.2 Light Field depth estimation

Depth estimation on light fields is an extremely rich and still active field of research. Starting with classical computer vision approaches, Yu et al. [157] analysed the geometric structure of 3D lines in a light field image and obtained depth maps by matching those lines between sub-aperture images (SAI). Tosic et al. [158] formulated a method to construct light field scale-depth spaces, indicating regions of constant depth, before solving the finer depth estimation in each space separately. This allowed to obtain good results in both highly textured and uniform regions. Zhang et al. [8] provided a solution to deal with occlusion artefacts, by implementing a spinning parallelogram operator to divide EPIs into regions and locating depth lines by maximising distribution distances of those regions.

After the advent of deep learning, several new methods were developed. Based on the EPI or epipolar geometry property, Luo et al. [159] proposed to formulate the depth estimation as a classification problem, in which a standard CNN-architecture is employed on horizontal and vertical EPI patches. Since a shallow CNN is inadequate to guarantee proper accuracy, a global optimisation with traditional approach is utilised. Feng et al. [160] presented a similar approach in which a shallower CNN is considered and the output of the fully-connected layer is more than one pixel. Jiang et al. [161] proposed to estimate initial depths by a fine-tuned flow-based network and then refine these initial results using a multi-view stereo refinement network. Shin et al [162] presented Epinet, an end-to-end network to predict depth, which takes as inputs the horizontal, vertical, left diagonal and right diagonal camera views, instead of EPI patches. With richer information of light fields, Epinet achieves a better accuracy. Khan et al. [163] used the idea that depth edges and more sensitive than texture edges to local constraints, and tell the two apart using a bidirectional diffusion process. Some of the most recent work is looking at using attention-based models, providing a better selection of features even is complex and texture-rich scenes, and leading to more accurate disparity (Tsai et al. [164]) or depth maps (Chen et al. [165]).

### 5.2.3 NeRF

The seminal paper by Mildenhall et al. [18] is intended to be a novel view synthesis method, and does so by rendering and optimising a continuous volumetric scene using a sparse set of input views. The input to their fully-connected non-convolutional network is a single continuous 5D coordinate (spatial location and viewing direction), which output the volume density and view-dependent emitted radiance at that location. While intended for novel view synthesis, as the network performs a dense 3D reconstruction of the scene, it can also be used for accurate high-resolution depth estimation, generated concurrently with novel views. Building on this foundation, and looking more specifically at the problem of depth estimation on indoor scenes, Wei et al. [166] combine structure-from-motion (SfM) and learning-based priors and plug them into a NeRF network to obtain high-resolution depth estimation. The sparse SfM reconstruction is fine-tuned using a monocular depth network, and use those priors to fix the inherent shape-radiance ambiguity of NeRF. Finally they further improve the results by using a per-pixel confidence map.

## 5.3 Comparing novel view synthesis

We first describe the data used in this section. We selected three images from the HCI synthetic dataset ($boardgames$, $rosemary$, $table$) [147], and five images from Lytro datasets: INRIA ($fruits$) [124], EPFL ($bikes$) [123], and our own ($guinness$, $frog$, $cards$), the latter two containing full camera calibration data [167]). These two types of images pose a challenge to NeRF because the principal point of these images is not centered, which is expected by NeRF. To counter this issue we modify the original SAIs to shift the focal plane to infinity. While this centres the principal view by simulating the images being taken by a single camera, it comes at the price of a small loss of resolution. In addition we use one image from the Stanford gantry dataset ($lego\ knights$) [12], and four high resolution images from the Technicolor ($birthday$, $painter$) [168] and SAUCE datasets ($cellist$, $fire\_dancer$) [45, 14]. These high resolution images, with a wider baseline, pose some difficulty for traditional light field methods which are not designed for such sets. The gantry and high resolution set, having been captured by a sin-

gle camera, do not suffer from the aforementioned issue, and NeRF can handle them directly.

### 5.3.1 Methods

There are many traditional methods for light field novel view synthesis, who were initially all depth-based, but some novel methods were proposed to increase the accuracy of the reconstruction. For example Vagharshakyan et al. [149] use the EPI representation of light fields, and perform inpainting in the shearlet transform domain to generate novel views between two existing views.

Those methods were soon replaced with machine learning approaches, the main drawback of which is the need for large amounts of labelled data to train any network. Chen et al. [153] bypass this issue and instead first train their network on labelled video data, more widely available, and use in turn a self-supervised network guided by a cycle consistency constraint, used to build bidirectional mapping and enforce the generated views to be consistent with input views.

On the other hand, NeRF works by approximating a continuous 5D scene representation with an MLP network, whose input is a 3D location $(x, y, z)$ and a 2D viewing direction $(\theta, \phi)$. Its output is an emitted colour $(r, g, b)$ and volume density $\sigma$. The weights obtained encode the volume of the underlying scene by mapping each input 5D coordinate to its volume density and emitted colour. This model is view dependent, which allows it to handle non-Lambertian effects and realistic specularity while rendering novel views.

### 5.3.2 Visual results

Looking first at large baseline images, NeRF seems to have offer better accuracy of reconstruction, see Figure 5.1. Several points must be noted. First of all, NeRF can work on the full resolution image, while most traditional light field methods work better with square (cropped) views, and as a result there is necessarily loss of information in the second case. On top of that, the difference in quality between the reconstructed views is

(a) (b) (c) (d)

Figure 5.1: View synthesis on image $birthday$ (detail) and $lego\ knights$ obtained using the method of [153] (a, c) and NeRF output (b, d).

pretty obvious, NeRF comes out with high quality and high resolution novel views, capturing most of the minute details of the scene, even in complex ones like in Figure 5.1 (a&b), despite the high number of occlusions present. Traditional light field methods however come out with noisier results, as if motion was present.

Comparison on smaller baseline images shows both approaches seem to have their issues, and it is more difficult to determine which is preferable, see Figure 5.1 (c&d). While traditional light field methods seem to work fairly well, on some depth levels it is still possible to see some reconstruction artefacts, as if motion occurred, however those issues are not generalised: notice how the shield in the corner has high detail, but the rest of the image suffers from artefacts akin to motion blur. NeRF on the other hand seems to handle some parts of the image fairly well (helmet, spear, wall), but suffers from reconstruction artefacts in many places which detracts from the details in the rest of the image. In particular those artefacts seem to occur on the edges of the image, however it is not limited to that (sword in the centre).

One thing to note, even though possibly obvious, is that both methods can only render novel views within the angular space defined by the input views.

### 5.3.3 Objective comparison

For objective comparison we use two classic metrics, PSNR and SSIM. They are properly representative as both traditional light field methods and NeRF generate views that are directly aligned with existing views, used

Table 5.1: Metric results (PSNR and SSIM) on novel views, comparing method of [153] and NeRF.

| | $P-LF$ | $P-NF$ | $S-LF$ | $S-NF$ |
|---|---|---|---|---|
| boardg | 34.83 | 43.13 | 0.912 | 0.993 |
| rosemary | 34.23 | 41.28 | 0.904 | 0.983 |
| table | 33.91 | 39.19 | 0.895 | 0.954 |
| E_bikes | 32.36 | 32.45 | 0.862 | 0.963 |
| I_fruits | 31.63 | 30.29 | 0.856 | 0.948 |
| V_guinn | 32.92 | 33.21 | 0.848 | 0.940 |
| V_cards | 32.47 | 33.17 | 0.861 | 0.957 |
| V_frog | 35.69 | 41.63 | 0.873 | 0.982 |
| legoK | 21.61 | 24.67 | 0.711 | 0.849 |
| birthday | 19.29 | 23.69 | 0.542 | 0.750 |
| painter | 23.24 | 28.08 | 0.563 | 0.786 |
| cellist | 27.18 | 35.70 | 0.632 | 0.970 |
| fire_danc | 27.25 | 30.82 | 0.625 | 0.972 |
| Mean | 29.74 | 33.64 | 0.714 | 0.927 |

as ground truth. As we can see in Tab. 5.1, NeRF performs better on all datasets. This is not surprising as it renders the scene in a continuous underlying 3D model, which is then used to generate novel views and thus does not have to approximate parts of the scene. Both methods seem to fare better with smaller baseline data, however when looking at wider baseline there is a large discrepancy between the images used. We posit that since the $birthday$ image is rife with minuscule details, it is more difficult to generate novel views that fool the metric well, even though it fools the eye, whereas the $cellist$ image contains a large uniform area and limited number of elements in the scene.

## 5.4 Comparing depth estimation

While both traditional light field methods and NeRF allow to obtain accurate results in their preferred environment, neither method truly generalises to all types of data. For this comparison, since we need ground truth depth estimation to properly use the selected metrics, the only data for which we have objective comparison are the synthetic images. For the others, visual comparison will be used instead.

Table 5.2: Metric comparison results (MSE + Bad Pixel Count) on depth estimation performed on synthetic images, between method of [9] and NeRF.

| | $M-LF$ | $M-NF$ | $B2-LF$ | $B2-NF$ |
|---|---|---|---|---|
| boardg | 4.513 | 0.602 | 15.41 | 3.57 |
| rosemary | 7.135 | 1.916 | 17.84 | 5.12 |
| table | 6.205 | 1.931 | 17.12 | 5.86 |
| Mean | 5.951 | 1.483 | 16.79 | 4.85 |

### 5.4.1 Methods

When it comes to traditional light fields methods we used here [9], and their process follows a three-step approach. First they extract a 3D volume of the light field by selecting views along a single angular dimension. Second they perform an optical flow estimation to obtain disparity estimates between the selected views. Finally the last aggregation step allows to obtain depth maps from the multiple disparity map estimates. This process is relatively fast and runs in about $20$ seconds for a whole row or column of the light field.

NeRF on the other hand is providing with 'direct' disparity estimation, which can then be converted to depth estimation, as the network first trains to obtain a 3D representation of the scene, from which each new view is rendered, as well as the corresponding depth. As a result the accuracy of the latter is very high, at the cost of higher computational time. For example, generating a single novel view of resolution $512x512$ takes an average of $18$ seconds, while the same for a view of resolution $2048x1080$ takes an average of $3$ minutes.

### 5.4.2 Visual comparison

From the images selected the differences are clear between both approaches. On high quality, high baseline images (see Figure 5.2) NeRF has a clear advantage. By generating a comprehensive 3D render of the scene, it has access to fine features and details from the scene. Considering that representation, it also has detailed information regarding camera position and its distance to every single point of the scene, which, at the same time as it allows the generation of high quality novel views, also helps generating a corresponding high quality disparity map. Some artefacts can be

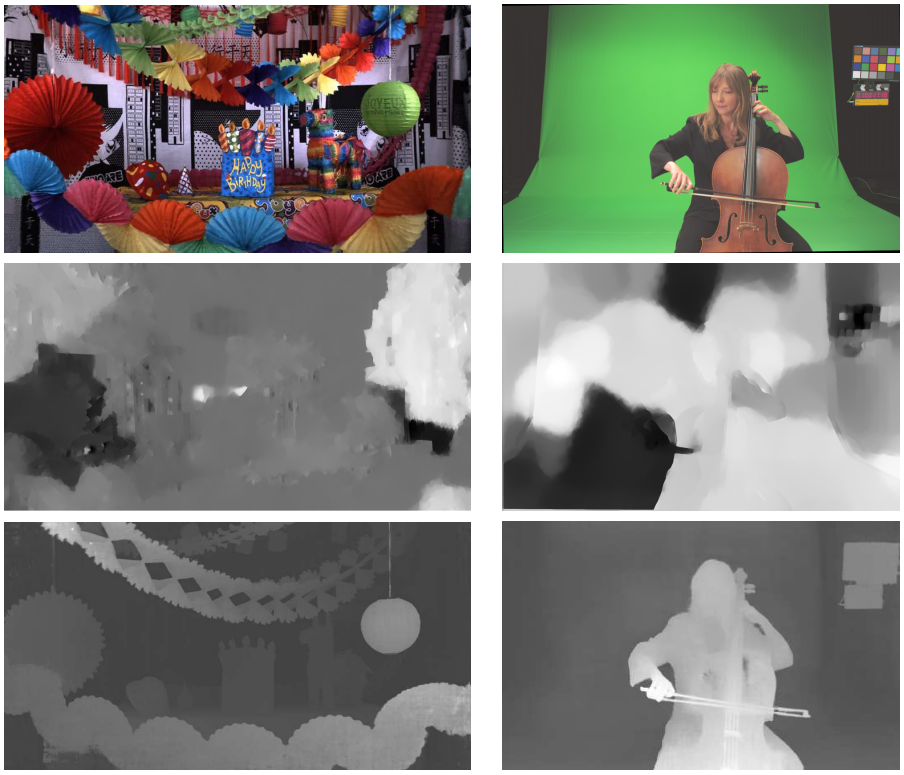


Figure 5.2: Depth maps obtained using method of [9] (middle) and NeRF output (bottom) on the $birthday$ (left) and $cellist$ (right) images.

visible on the edges of the image, which we posit could be explained by the lower number of views in which those parts are present, which lowers the quality of the rendering in these parts. Traditional light field methods on the other hand fail to accurately obtain a proper depth estimation, in part due to the higher baseline, the absence of camera parameters, and the fact that the input views are not aligned on a perfect grid - as the data we used was unstructured - which is one of the expectations those methods have.

When it comes to smaller baseline images, the advantage is still on the side of NeRF, however there are counter-examples, see Figure 5.3. We posit this image in particular is tricky for NeRF as only parts of small objects come into view on the sides of the images and potentially make a 3D representation of the scene, this already led to artefacts visible in Figure 5.1.

Figure 5.3: Depth maps obtained using method of [9] (centre) and NeRF output (right) on $Lego\ Knights$ image.

### 5.4.3 Objective comparison

For comparison on depth estimation, we use again two metrics, MSE and Bad Pixel Count (2.0), see Table 5.2. Comparison for synthetic images is fairly straightforward as ground truth exists, and in this case we notice that NeRF does perform much better compared to traditional methods. However when it comes to comparing natural images, whether Lytro or otherwise, ground truth depth maps do not exist, making objective comparisons less relevant. We therefore only use visual comparison for these images.

## 5.5 Conclusion

We have presented in this chapter a comparative study between the newly developed NeRF with regards to view synthesis and depth estimation, and traditional light field methods, which it aimed to replace. While very impressive, NeRF has shown a minor limitation in the data type it can process directly, and tricks need to be used to make it usable with either synthetic or Lytro image. In general, we can say that both schemes have their advantages for the specific type of data they target, but do not directly generalise to any type of light field, which leaves traditional light field research some opportunities for higher quality applications. Some possible future work related to this chapter include adapting some editing methods that use traditional light fields to using NeRF instead, and analysing the quality of those results. In particular using NeRF for the work presented in Chapter 4 would be of particular interest.

# Chapter 6

# Conclusion

We approach the end of this dissertation, where we looked at the possibilities offered by light field for high quality editing. We take in this chapter some time to reflect on the contributions made in this thesis through a summary of the work, and follow by considering the future applications of this work, as well as the avenues opened by it.

## 6.1 Summary

In Chapter 1, we presented light fields and discussed the vast array of applications available, before shifting our focus on editing. We showed that light fields data are far from perfect and suffer from a number of drawbacks, most of which stem from physical and technical constraints that are built-in the capture devices. Nevertheless we explain that this data is important and deserves to be restored and enhanced.

In Chapter 2, we went on a journey through the history of light fields, from ancient to current days. We also formalised light fields through defining the plenoptic function, and the concept of light field rendering. We described in detail the different methods of capture and the advantages each bring, as well as their limitations. We looked beyond the theoretical work that light fields offer, and discussed a few of the most prominent fields of application it offers. Finally we stopped for a moment on the most modern technique to generate light fields, and the large body of research

it created.

In Chapter 3 we presented our solution to restore and enhance the quality of Lytro images. We showed that even through software applications it is possible to mitigate many of the hardware-generated artefacts inherent to this type of camera, namely distortions, brightness and colour balance, and noise. We also showed, using both objective and subjective analysis, that our solution performs better than the previous state of the art, showing that every part of our pipeline is necessary for optimal effect. Additionally we provide a number of examples of typical light field applications performing better using our enhanced data.

In Chapter 4 we took the improved Lytro images we produced in the previous chapter, and used them to apply soft colour segmentation, with the intent on using the layers for colour editing. This was motivated both by the idea of performing better colour segmentation using light field data, notably to account for occlusions, but also to perform better editing by separating the objects in the colour layers using depth cues. We showed that these tasks perform adequately on most images, and discussed some limitations, specifically regarding non-Lambertian objects.

In Chapter 5 we looked at a new promising technology, NeRF, and attempted to compare it with traditional methods in the applications of view synthesis and depth estimation, in which it excels. Nevertheless we put this in the broader context and expressed limitations regarding the type of light field data it can process, or the need to manually transform certain data into becoming compatible with it, and the computational power and time it requires. We show that it is still a very promising venue for the future of light field rendering and its many applications.

## 6.2 Outlook and Future Work

Research Question Revisited

*We attempted to investigate* — "***How can we benefit from Light Field images to perform high quality processing and editing?***".

Three main objectives are explored in this context:

- Lytro Image Quality Enhancement.
- Soft Colour Segmentation on Light Fields.
- Expanding the range of usable data using NeRF.

### On the topic of light field for editing

Throughout this thesis, we have attempted to use widely available light field data to perform a series of tasks, among which editing, and the results we showcased are quite satisfactory. While we have accomplished our initial goal to rid original plenoptic data of most of its defects, we find ourselves with low resolution images having a very small baseline, capturing mostly static scenes. This was a necessary first step toward better acceptance of light fields as a medium to perform editing, and this effort will need to be explored further.

While Lytro images could be replaced by other types in the future, we still learned valuable lessons by using them and they certainly benefited light field in immeasurable ways. Looking back at our contributions, the methods we provide for denoising, clearing sensor-induced hot pixel noise, or fixing colour imbalance between light field views, are all very relevant as we move on to using more advanced, higher quality images captured from contemporary rigs. Camera calibration is a very sensitive topic, and has still not been fully solved. Even with the most advanced setup, using two different models of the same camera is going to yield two different photographs. Perhaps this difference is near-imperceptible to the eye, but if these imperfections are spread on large camera arrays, and their output used on very colour-sensitive applications, it is likely to lead to processing artefacts. These need to be corrected before moving on to the editing tasks, and the methods we proposed in Chapter 3 will still be rel-

evant provided they are properly adapted.

Similarly, we presented in Chapter 4 methods that can easily be transposed to be used on higher quality data, captured not just from Lytro cameras, but any type of light field capture setup. Using depth information for segmenting objects should become the obvious solution when performing fine editing using light field data. As well, even if soft colour segmentation was used only as an example in our case, the benefit of having light field data of a scene for solving occlusions is invaluable and could be translated to many other editing tasks.

### On the future of light fields in industry

Widespread adhesion to light field imagery in under way in a number of fields of application, and a number of avenues for use are opening, in movie post-production owing to the pioneering work described in Section 2.3 by the groups engaged in the SAUCE project [45, 14] or in virtual reality by Huang et al. [52], among others. However some limitations still exist; the setups are in general cumbersome and not very portable, require large amounts of additional calibration to work optimally, and data storage and processing is made challenging by the sheer amount generated by the most recent light field setups. For many, these are going to be overwhelming drawbacks and they will prefer to use more traditional setups instead.

However, as is very often the case in computer science, these issues may only be temporary. Computational power and storage abilities are only going to grow, and it is possible than even within the next few years, light field solutions will become affordable and attractive to some production studios. Seeing the possibilities for image processing offered by light fields, it is almost certain that we will see them used more in the future.

## 6.2.1 Future work in the short term

Taking a step back to look at our own work, there are clear advantages brought out by the work presented in this thesis. However, there is still room for further improvements. For instance, some lens-induced distor-

tion are still visible on some of the outer Lytro sub-aperture views, and could benefit from being corrected. However, do they really need to be? It is possible that, as time goes, low resolution small baseline light fields are going to become less common. This makes solving these artefacts interesting for theoretical research, but perhaps less so for industrial or popular applications.

Similarly, on the subject of editing, the scope for future work is still wide. We showed in Chapter 4 that the computation time of our methods is still very high, despite processing low resolution images. This cost is mostly the product of the sheer amount of images representing a single light field, and this amount of data can only go up as technology evolves. Therefore it is important to find solutions to these software limitations, potentially by initialising decomposition of sub-views using previously processed results from their neighbours, thereby reaching convergence to an optimal solution faster.

Additionally it will be necessary to refine our object-based layer separation method. It is currently still very coarse and would need to be reformulated when working with more challenging scenes full of small objects and details. Finally these last two contributions could be studied in the temporal domain. If it is possible to easily propagate edits spatially along the light field, the process should also work between video frames, provided there is no cut in the shot and the main scene retains its colour properties, i.e. no object of unknown colour entering the scene, or one leaving it.

Lastly regarding the use of NeRF for light field editing, we need to come up with smart solutions to perform our soft colour segmentation. Following the example of Zhang et al. [77], it should be possible to plugin into their framework a module dedicated to learning how to compute soft colour segmentation, and obtain NeRFs in which the colour information can be split easily to create renders of individual layers. This could allow to directly generate edited novel views, only by editing the underlying model once.

### 6.2.2 Future work in the long term

It is our opinion that in the future the focus will shift from low-resolution plenoptic images towards using higher quality images, as this movement is already well under way. Learning to deal with the additional amount of data is going to be one of the main challenges to future light field processing, and while our methods can be adapted for those, the solutions for this are outside the scope of this thesis.

It would also be interesting to see a study comparing the output of the more advanced derivations of NeRF networks against some of the high-end light field camera arrays used nowadays in movie post-production. The ease of capture in the first case is a major advantage, but if the output finds itself being limited in its resolution and detail quality by the need to produce views through a machine-learning based render, perhaps it is not the most desirable method for high-end movie productions, although it could be appealing to regular users and photographers interested in expanding their processing options.

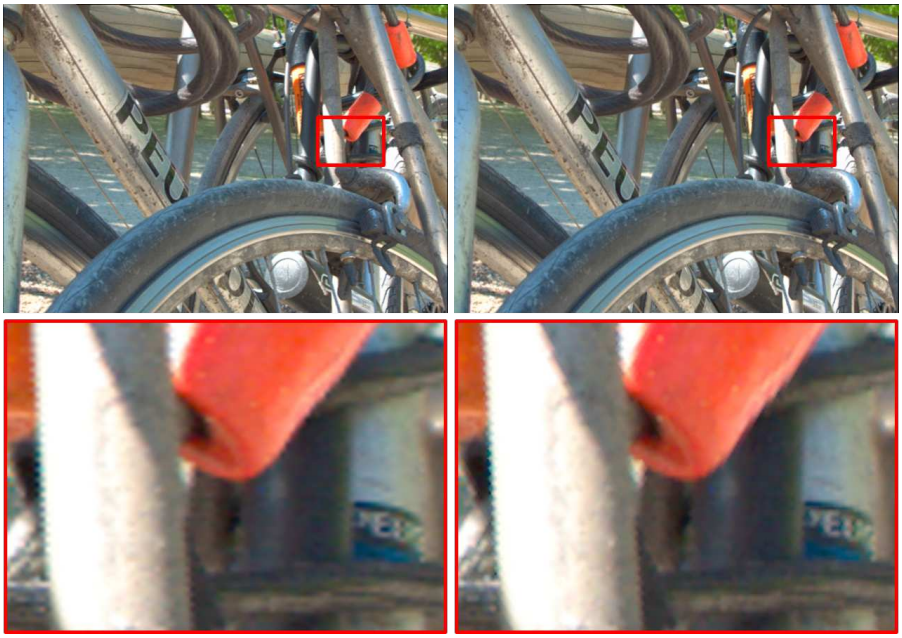
The work presented here is a small but useful iteration in a broad field of research. We have showcased some advances in light field processing and editing, and we hope that our methods will prove useful to future explorations.

# Appendix A

# Review of other light field extraction methods

## A.1 Barycentric interpolation [1]

Because of the hexagonal lenslet pattern, the conversion step from the lenslet image to sub-aperture images must include a resampling of each SAI from a hexagonal to a square pixel grid. For this purpose, the light field toolbox of Dansereau et al. [5] performs a fast 1D interpolation in each row of each SAI. Instead, the demultiplexing method in [1] introduces a barycentric interpolation method that produces higher resolution images (by a factor ~2.6). Figure A.1 presents a comparison of our demultiplexing using either the original method (1D interpolation from [5]) or the barycentric interpolation in [1]. For the comparison, bicubic upsampling was applied to our original demultiplexing. Although the difference is subtle, the barycentric interpolation slightly reduces the resampling artefacts. However it also multiplies the number of pixels by ~6.75 which would significantly increase the complexity of the next steps of the pipeline. Hence, we have kept the 1D interpolation from [5] in our paper. Furthermore, modern light field super-resolution techniques (e.g. [134]) significantly outperform the bicubic upsampling used in Figure A.1 (a), which is expected to further reduce the advantage of the barycentric interpolation.

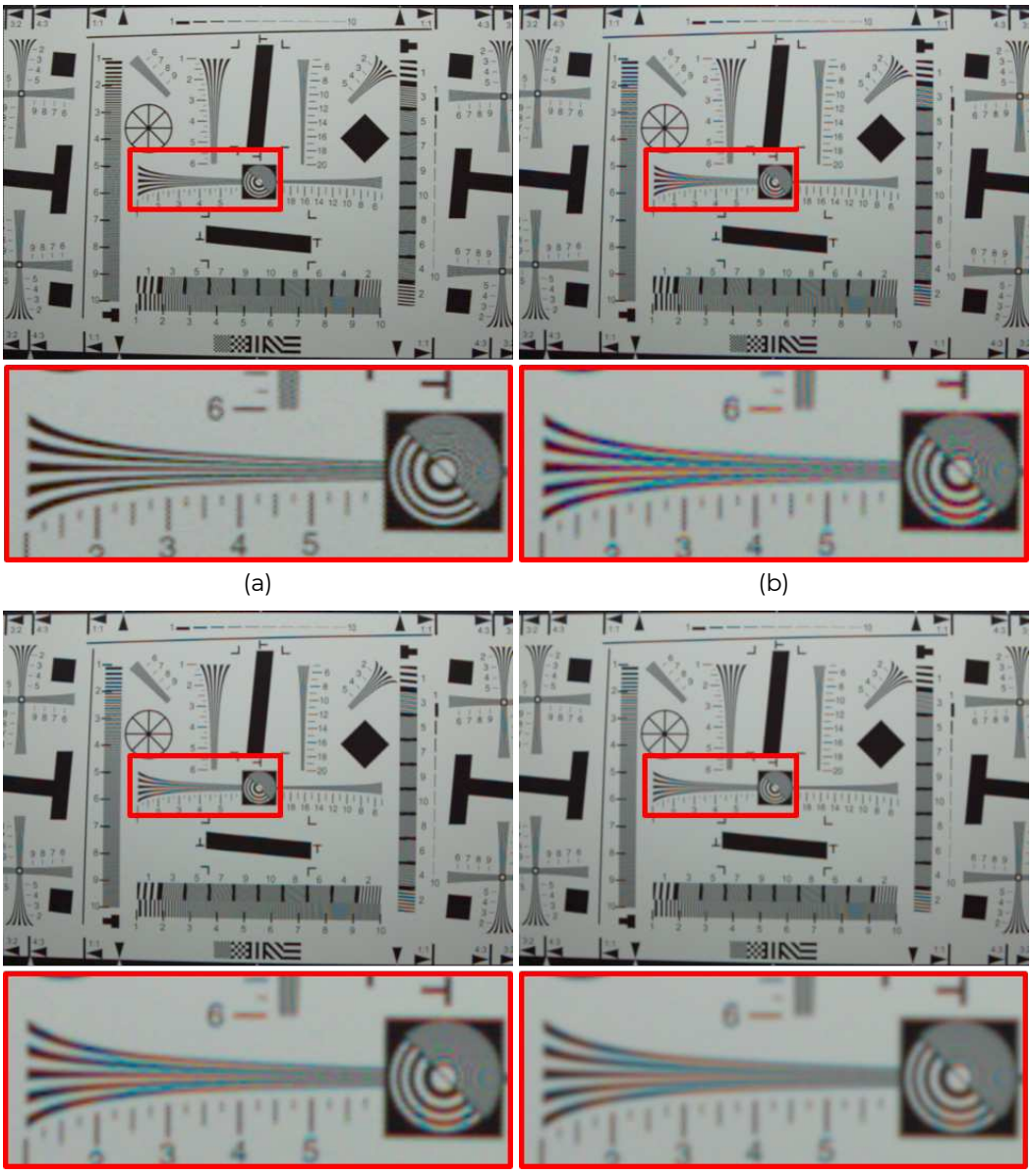


(a) (b)

Figure A.1: Light field *Bikes* demultiplexed with (a) our original demultiplexing (followed by bicubic upsampling), (b) our modified demultiplexing using barycentric interpolation [1] for the hexagonal to square SAI resampling.

## A.2 Demosaicing based on 4D Kernel Regression [2]

The demultiplexing approach in [2] uses kernel regression in the 4D light field space in order to perform the demosaicing instead of applying a 2D demosaicing of the RAW image. The method simultaneously performs the demosaicing with the other interpolation step of the pipeline (i.e. lenslet image rotation and hexagonal to square resampling). However, unlike traditional demosaicing methods, the RGB colour components are processed separately, hence the correlations between components are not exploited. For the comparison with our method, we have implemented the 4D Kernel Regression demosaicing within our pipeline. The results are presented in Figure A.2. Using a small kernel produces strong colour artefacts. Although these artefacts are reduced with larger kernel sizes, they remain more visible than with the 2D demosaicing in Figure A.2(a), and the result is blurred.



(a) (b)

(c) (d)

Figure A.2: Demultiplexing results with (a) our original demultiplexing (using 2D demosaicing from [111]), (b, c, d) our modified demultiplexing using 4D Kernel Regression demosaicing [2] with a gaussian kernel of standard deviation of respectively $\sigma = 0.45$, $\sigma = 0.6$, $\sigma = 0.8$.

(a) (b)

Figure A.3: Demultiplexing results where the lenslet array alignment step is performed using (a) bilinear interpolation, (b) nearest interpolation. In both cases, 2D demosaicing was performed before the alignment.

## A.3 Demosaicing based on disparity estimation [3]

The authors in [3] propose a different demultiplexing approach where sub-aperture images are extracted before applying demosaicing. This results in a set of mosaiced images with an irregular colour pattern instead of the traditional Bayer pattern. Disparity maps estimated from the incomplete SAIs are then used to fill the missing colour components of the pixels in each SAI with known pixel's component in other SAIs. However, the issue of performing the demosaicing at the end of the process is that the misalignment between the microlens array and the sensor cannot be compensated using interpolations with sub-pixel accuracy, since the colour data is incomplete. Hence only interpolation to the nearest pixel is applied. This may cause aliasing artefacts, especially in the regions that are not in focus in the original capture (i.e. with high frequencies within each lenslet, which corresponds to an angular patch). The effect of using nearest interpolation is shown in Figure A.3.

## A.4 Plenopticam software [4]

Similarly to the Light Field toolbox of Dansereau et al. [5], Plenopticam is a publicly available software that provides a complete pipeline for extracting the light field views from plenoptic camera RAW data. Figure A.4 shows an example of result using the current version of Plenopticam[1]. Note that the viewpoints extracted with the Plenopticam and our method may slightly differ. Hence, for a fair comparison of external views, we have selected the view extracted with our approach (Figure A.4 (c)) that is the closest to the external view shown for Plenopticam in Figure A.4 (d). Similarly to our method, the views extracted with Plenopticam keep globally consistent colours. However, their colour consistency processing reveals strong artifacts on the external views. Furthermore, the results have exaggerated colour saturation and contrasts compared to the reference image from the Lytro Desktop proprietary software. Note that gamma correction is already applied, but is performed directly on the RAW data, before devignetting. This may cause inaccurate devignetting further explaining the artifacts on external views. Finally, some details are lost in the highlights. On the other hand, our method recovers these details thanks to the highlight processing step.

[1] accessed from `https://github.com/hahnec/plenopticam` on the 09/12/2019.

Figure A.4: Comparisons of our method (including post-processing) in (a) and (c) with the Plenopticam results in (b) and (d). For (a) and (b), the extracted central view is shown above the red line, and a refocused image with Lytro Desktop proprietary software is shown below the red line to indicate the reference colours. The images in (c), (d) correspond to an external view of the light field.

# Appendix B

# Study of colour inconsistencies

While the vignetting phenomenon reduces the brightness on the borders of each lenslet, it should not affect the chromatic information in theory. However, in order to compensate for the microlens vignetting effect, the devignetting step applies a large gain to the pixels on the border of a lenslet, thus increasing any possible source of error for external SAIs. This results in more noise (see Section D), but also also causes colour inconsistencies in the light field.

We show in Figure B.1 that the colours of external SAIs essentially depend on the order in which the devignetting and demosaicing steps are performed. Unlike our approach in Figure B.1(a), no colour loss is observed when the devignetting step is performed after the demosaicing (see Figure B.1 (b) and (c)). Note that in this case, more reliable colours are obtained when the demosaicing is applied to both the RAW image and the White Image used for the devignetting as shown in Figure B.1(c). Demosaicing the white image results in a slightly coloured signal that compensates for some colours errors (e.g. green colour on the top-corner and red tones on the left side of Figure B.1(b)). These errors can be explained by the fact that the colour responses of the red green and blue pixels on the Bayer filter array are not perfectly uniform over the sensor.


Figure B.1: Effect of the order of the devignetting and demosaicing steps on the demultiplexing for an external SAI: (a) devignetting first (i.e. proposed approach), (b) devignetting after demosaicing of the RAW image, (c) devignetting after demosaicing of both the RAW image and the White Image, (d) centre SAI with intended colours (identical for each variant).

However, because of the high frequency vignetting pattern, performing the demosaicing before the devignetting also results in stronger high frequency artefacts (e.g. horizontal lines in Figure B.1 (b) and (c)). Therefore, we have preferred the approach in which the devignetting is applied first. Although the demosaicing causes a loss of colours on external SAIs in this case, this can be corrected with our post-processing recolouring step, unlike the artefacts in Figure B.1(c).

# Appendix C

# Results on Stanford dataset

In this section we show some results of objective metric evaluation for 10 light fields from the new Stanford 3-view dataset [125]. For this experiment 8 light fields were selected from the outdoor captures and 2 from the indoor captures. Although each capture consists of 3 light fields taken with a rig of 3 Lytro cameras, we have only used the central one here. The selected light fields are referenced in the results by their number as given in the dataset of RAW light fields.

### Colour consistency

In this section, we present the results of the colour consistency metrics on the selected light fields from the Stanford dataset. In Figure C.1 we present the results of the PSNR, SSIM, S-CIELab and histogram distance metrics, similar to Figure 3.10 in Chapter 3. We compare the results of Dansereau et al. (*Da*) with our decoding results (*De*) and the results we get after recolouring (*Re*).

Similar to the paper, we can see that (*Re*) performs the best with respect to PSNR, SSIM and S-CIELab, followed by (*De*) and (*Da*). Again, the histogram distance metric is showing some cases in which our algorithm does not remove all of the colour consistencies that appear across the light field. We have also included visual results in Figure C.2, which highlight that although the recolouring fixes the majority of colour changes to the outside

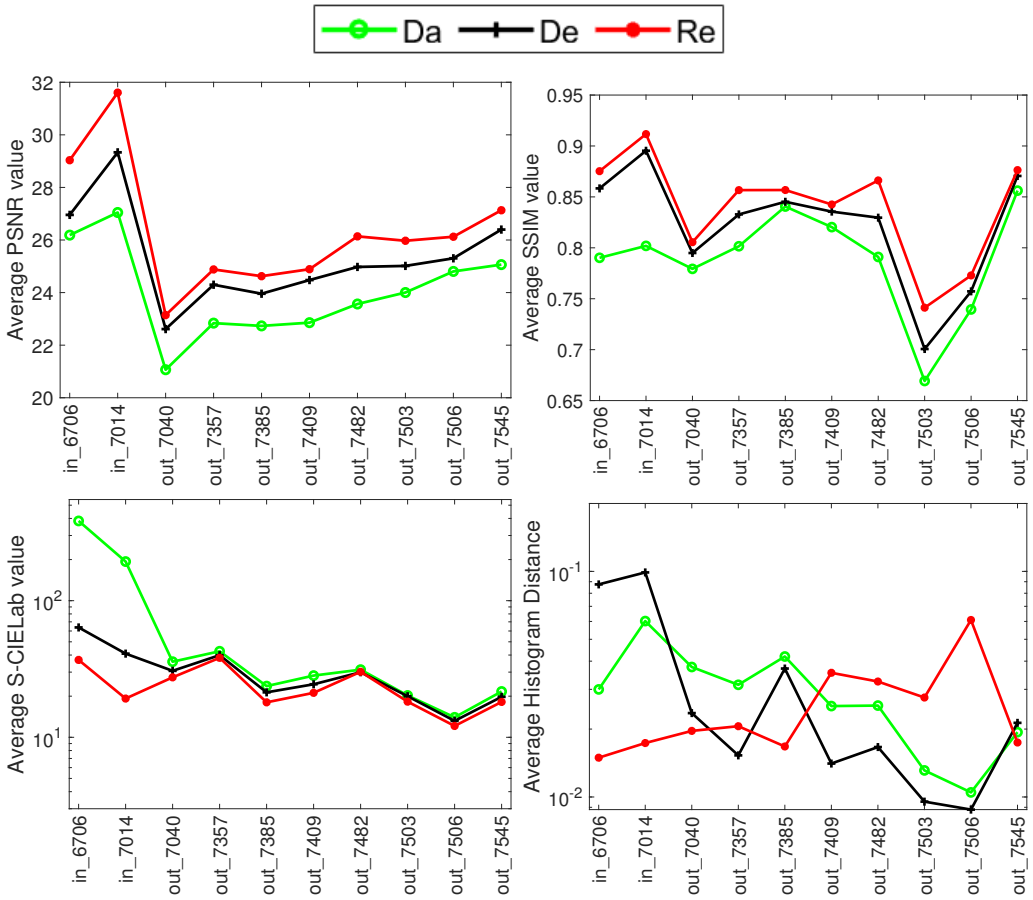



Figure C.1: Metric comparison, using PSNR, SSIM [126], S-CIELab [127] and histogram distance. Higher values are better in terms of PSNR and SSIM, and lower are better for S-CIELab and the histogram distance.

SAIs, some small differences remain. For example, in Figure C.2(b), small colour differences are still visible in the sky between the recoloured outside and centre SAIs, although a lot less visible than Figure C.2(a) before recolouring. In Figure C.2(f) the blue rectangle hasn't been completely corrected. This is due to the nature of our thin plate spline transfer function, which finds the best global, smooth transfer function to fix the colours in the image, and while this prevents severe artefacts changing the structure of the image, small local differences between the centre and outside SAIs can remain after recolouring. This is discussed in detail in Section VII-A of the paper.

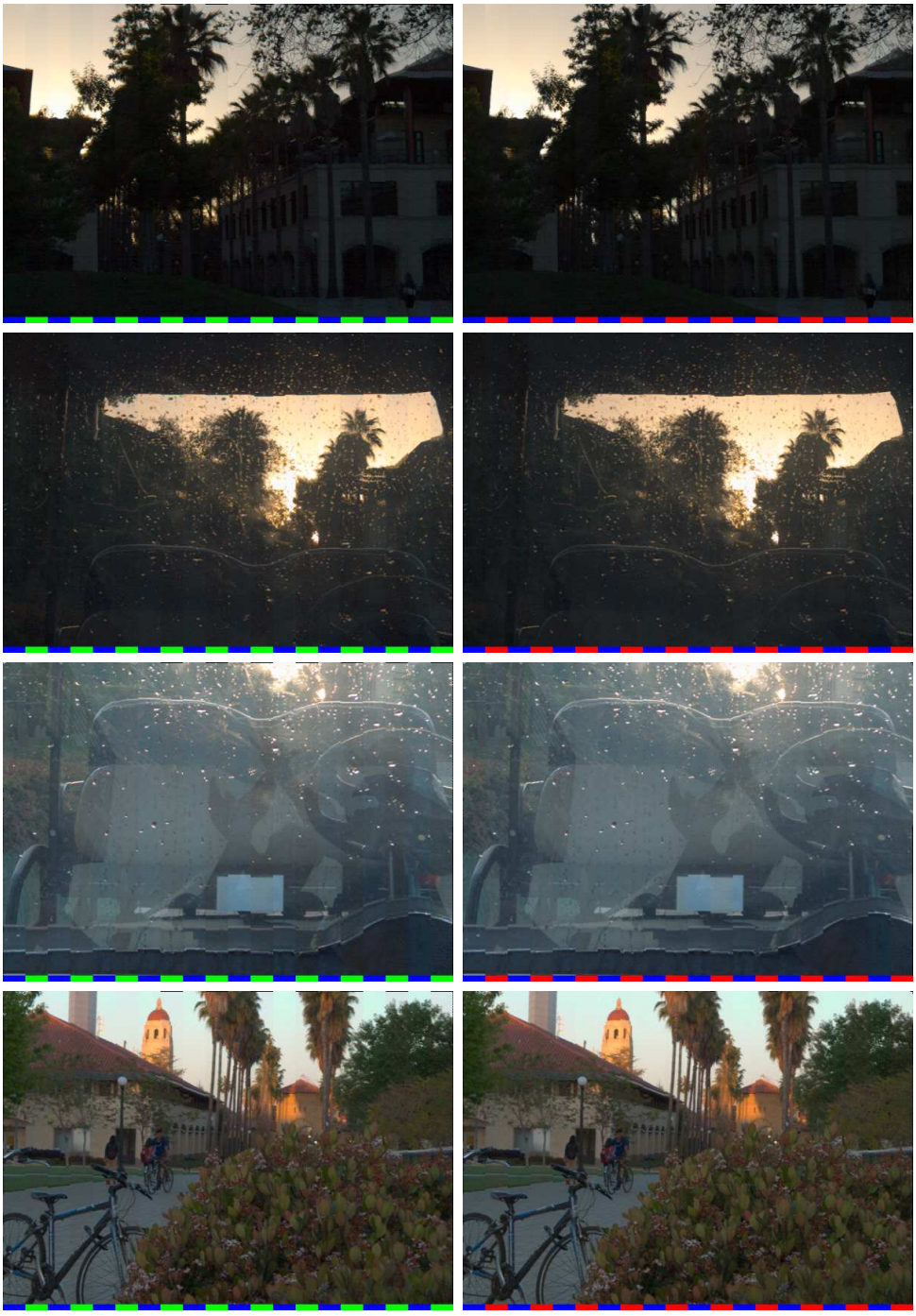


Figure C.2: Recolouring results for the light fields out_7482, out_7503, out_7506, and out_7545 (from top to bottom). The centre SAI is overlaid in column blocks onto one of the outside SAIs before recolouring (left) and after recolouring (right). The colours at the bottom of the images indicate which SAIs the columns are taken from - the centre SAI (blue), the outside SAI before recolouring (green) or the outside SAI after recolouring (red). Zoom in for more clarity.

Table C.1: Noise level $\sigma_{est}$ estimated using [128] for the new Stanford dataset [125] and each setting combination described in Table I of the main paper. The 3 setting combinations including denoising are shown on the right.

| $\sigma_{est}$ | $Da$ | $De$ | $DeH$ | $Re$ |
|---|---|---|---|---|
| in_6706 | 3.84 | 2.67 | 2.65 | **2.26** |
| in_7014 | 3.33 | 2.20 | 2.19 | **1.86** |
| out_7040 | 6.49 | 5.68 | 5.66 | **5.21** |
| out_7357 | 4.42 | 3.74 | 3.71 | **3.49** |
| out_7385 | 5.85 | 5.15 | 5.14 | **4.74** |
| out_7409 | 4.33 | 3.76 | 3.75 | **3.46** |
| out_7482 | 4.70 | 4.06 | 4.05 | **3.80** |
| out_7503 | 3.70 | 3.26 | 3.25 | **2.66** |
| out_7506 | 2.50 | 2.31 | 2.29 | **1.93** |
| out_7545 | 3.47 | 2.99 | 2.96 | **2.78** |
| Average | 4.26 | 3.58 | 3.56 | **3.22** |

### Noise level estimation

Here we estimate the noise level after each step of the pipeline using the blind metric. As for the other datasets, the noise is estimated on each individual SAI and the averaged result gives a score for a light field. These can be seen in Table C.1. A similar analysis can be done compared to the results of the other data. The demultiplexing step improves slightly on the output of Dansereau et al. [5], the hot pixel removal tool has little impact as hot pixel noise differs from the AWGN. The recolouring step again lowers the noise slightly.

# Appendix D

# Analysis of the noise profile

To perform a more in depth analysis of the noise of light fields captured with the Lytro Illum camera, we created a noisy light field dataset consisting of 5 scenes, shown in the top row of Figures D.2 to D.6. For each scene, 3 different noise levels were created by setting the ISO to 80, 250, and 640, The shutter speed was then manually adjusted so that the image is as bright as possible without saturation, using the Lytro Illum built-in real time saturation detection. For each scene and ISO setting, ~30 noisy instances were captured. We use the Genie Mini[1] rotating platform to automatically trigger the capture, without having to physically press the camera trigger, which could cause misalignment issues (the camera was not mounted on the platform). However, the automatic trigger sometime failed, resulting in 29 instances instead of 30 for a few scenes and ISO settings. While we used a white backdrop in the scenes, the dataset was captured in a green screen studio with stable LED lights, which ensured stable lighting conditions. The setup used is shown in Figure D.1.

A ground truth noise free light field was then created for every scene and ISO setting by averaging the noisy instances. For validation purpose we also created noise free light fields by computing the median of the noisy instances, which was found to be very close to the mean light field, at least ensuring the distribution of the noise is not skewed. The light field noise can then be obtained by removing the noise free light field from

[1] https://syrp.co/discovery-product/genie-mini/

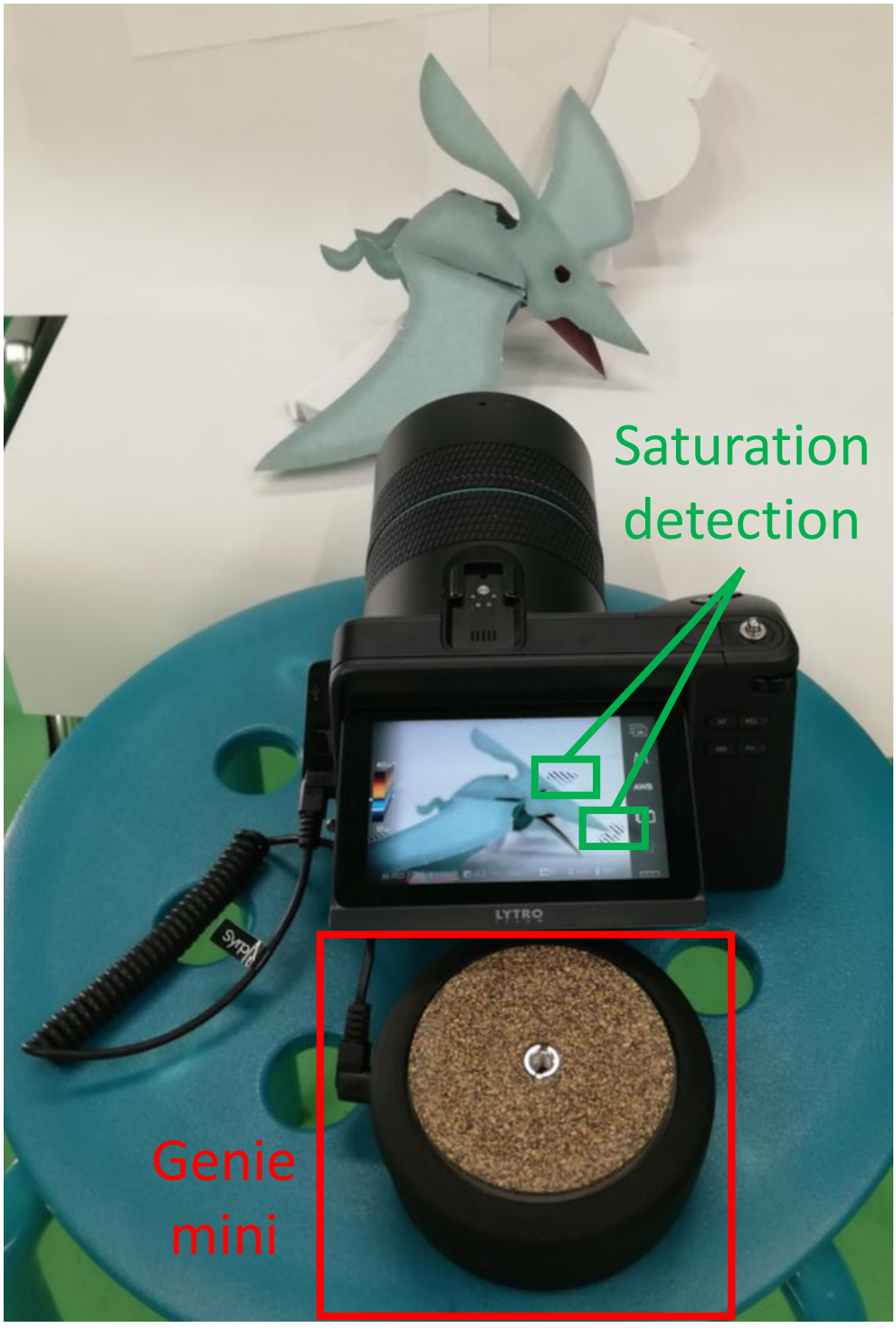



Figure D.1: Setup used to create the noisy light field dataset. The Genie Mini rotating platform seen next to the Lytro Illum camera was used to automatically trigger the shots. For each ISO setting, shutter speed was adjusted to maximise image brightness while avoiding saturation.

the noisy instances. For each scene and ISO setting, we then analysed the noise histogram per SAI. We show in Figures D.2 to D.6, noise histograms for the less noisy (left) and noisiest (right) SAI, together with a fitted normal distribution. The results show that the AWGN model is valid for each SAI, but SAIs of a same light field can exhibit different noise level. We used the standard deviation of the fitted normal distribution as a ground truth for the noise level per SAI. The last row of Figures D.2 to D.6 shows the noise level per SAI, normalised over the colour channels and the 3 ISO settings. As expected the overall noise level increases with the ISO, and we can clearly observe that the noise level is higher for outer SAIs due to vignetting.

Finally, we evaluated the blind metric used in the paper [128] by comparing the estimated noise level to the ground truth. Figure D.7 shows the graphs of estimated noise level, averaged over all SAIs, against the ground truth noise level for each scene. The last graph on bottom right shows the results averaged over all light fields. While the blind metric does not evaluate the exact noise level, a near linear relationship between the ground truth and estimated noise level can be observed, which validates the use of the chosen metric for the evaluation of our pipeline.

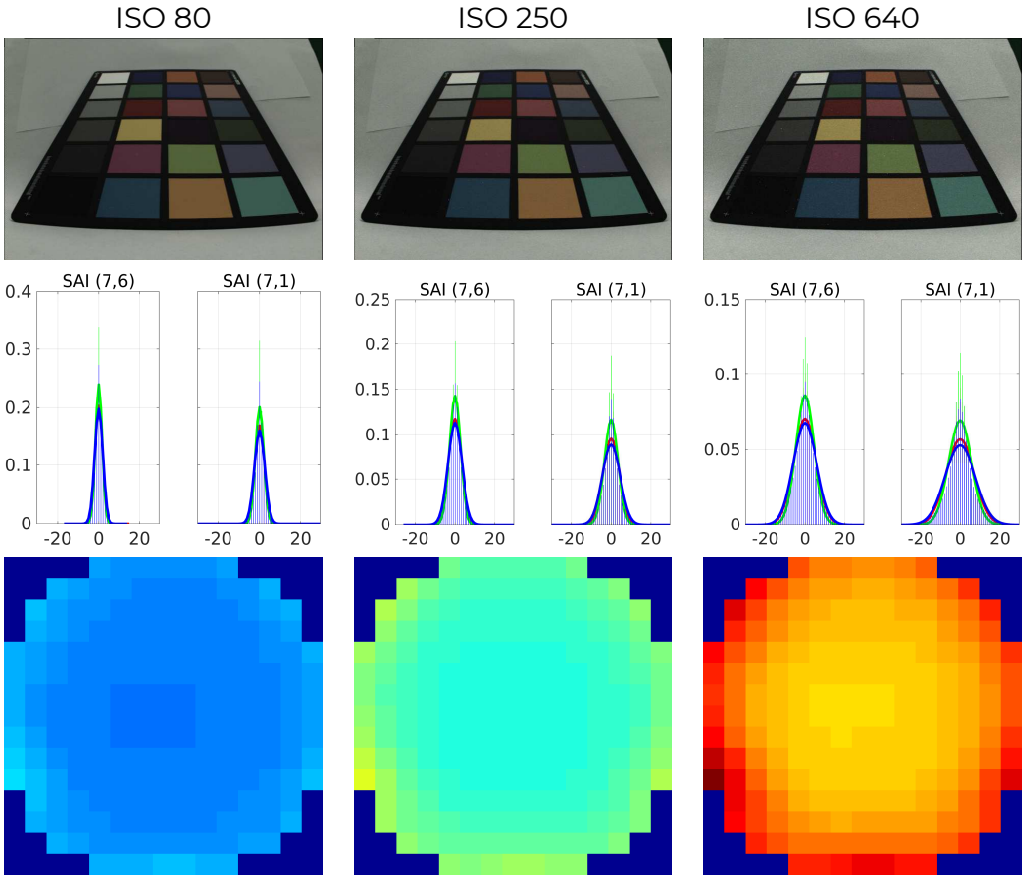



Figure D.2: Light field *color_chart* from the new noisy light field dataset. On top row, one of the 30 instance of noisy light field captured, noise increasing with ISO gain from left to right. On middle row we show the histogram of the less (left) and most (right) noisy SAI, together with the fitted normal distribution. On bottom row, we show the noise level per SAI, normalised over the colour channels and the 3 ISO settings.

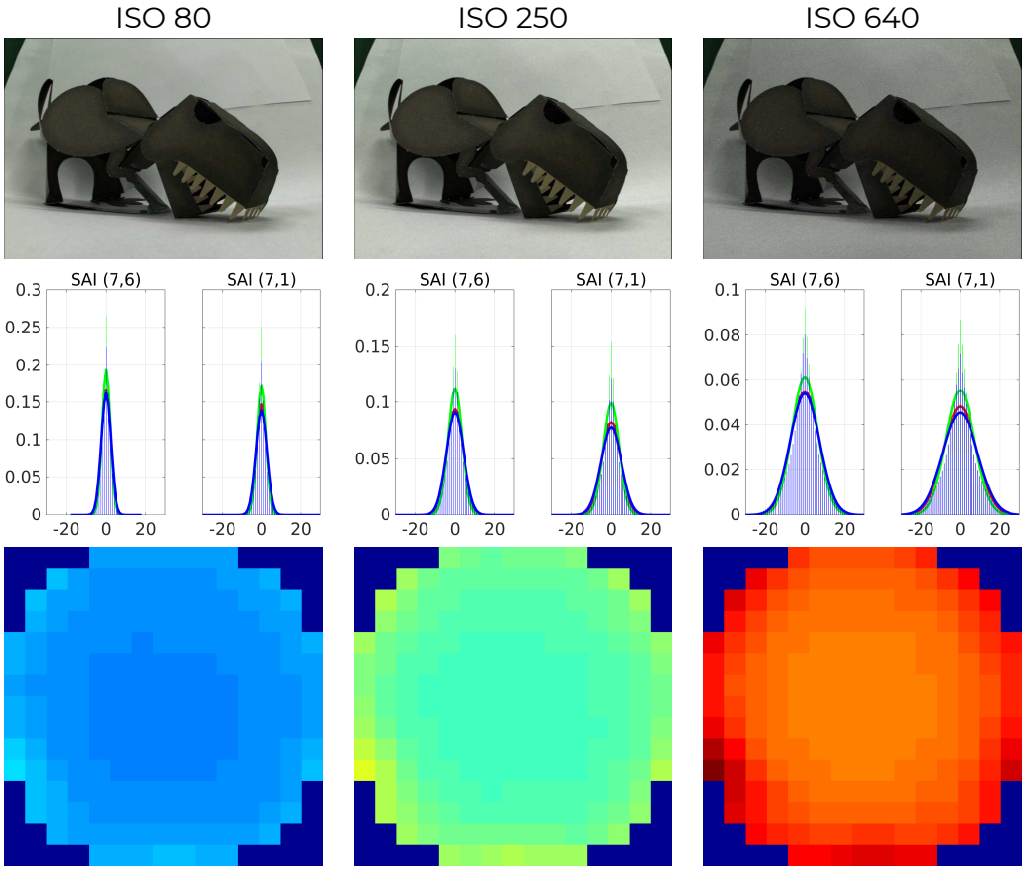



Figure D.3: Light field *godzi* from the new noisy light field dataset. On top row, one of the 30 instance of noisy light field captured, noise increasing with ISO gain from left to right. On middle row we show the histogram of the less (left) and most (right) noisy SAI, together with the fitted normal distribution. On bottom row, we show the noise level per SAI, normalised over the colour channels and the 3 ISO settings.

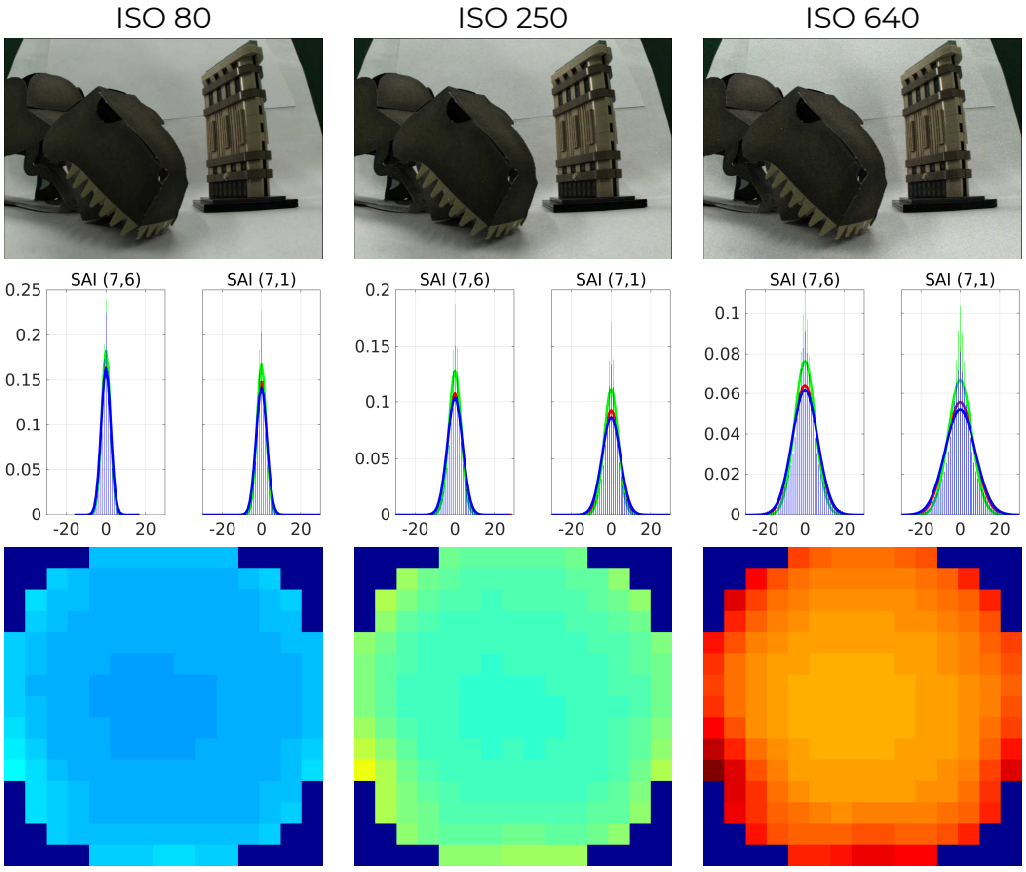



Figure D.4: Light field *godzi and lego building* from the new noisy light field dataset. On top row, one of the 30 instance of noisy light field captured, noise increasing with ISO gain from left to right. On middle row we show the histogram of the less (left) and most (right) noisy SAI, together with the fitted normal distribution. On bottom row, we show the noise level per SAI, normalised over the colour channels and the 3 ISO settings.

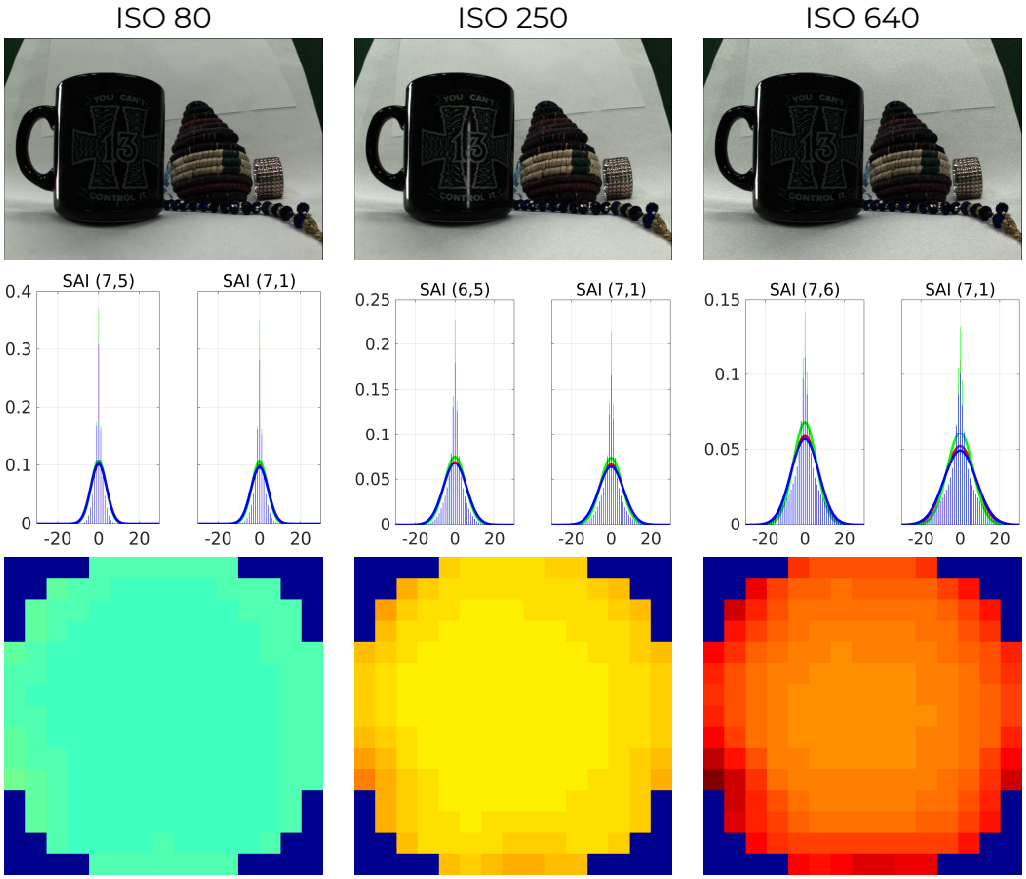



Figure D.5: Light field *mug* from the new noisy light field dataset. On top row, one of the 30 instance of noisy light field captured, noise increasing with ISO gain from left to right. On middle row we show the histogram of the less (left) and most (right) noisy SAI, together with the fitted normal distribution. On bottom row, we show the noise level per SAI, normalised over the colour channels and the 3 ISO settings.

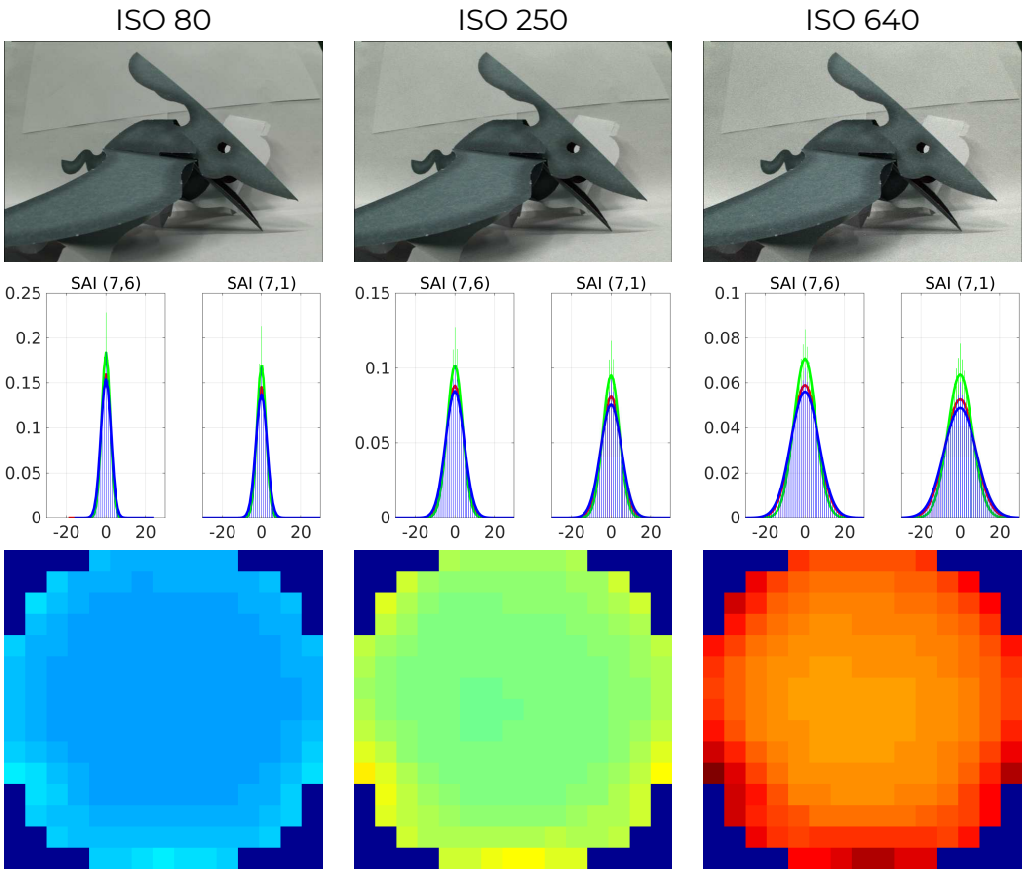



Figure D.6: Light field *polly* from the new noisy light field dataset. On top row, one of the 30 instance of noisy light field captured, noise increasing with ISO gain from left to right. On middle row we show the histogram of the less (left) and most (right) noisy SAI, together with the fitted normal distribution. On bottom row, we show the noise level per SAI, normalised over the colour channels and the 3 ISO settings.

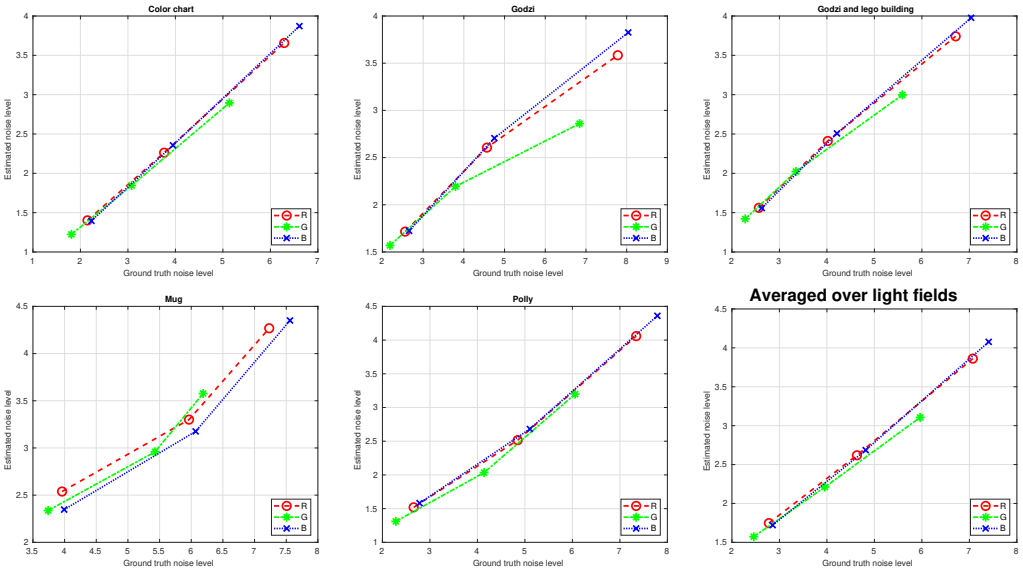



Figure D.7: Blind noise level estimation [128] plotted against the ground truth noise level.

# Appendix E

# Disparity / depth estimation

We evaluate here the performance of the proposed pipeline on depth or disparity estimation, which is one of the flagship applications for light fields. For that purpose we use 4 different methods [6][7][8][9] applied after every step of the pipeline. For all methods we used the code provided by the authors. The first method estimated the depth by simply computing the slopes of the EPIs based on the light field gradient [6]. Note that we used the code provided by the authors which implements the first step described in the paper and only outputs a sparse estimation. The second method was designed to be robust to occlusions by analysing the statistics of angular patches of the light field together with refocus cues [7]. The third method uses the spinning parallelogram operator to estimate the slopes of the EPIs and provide a robust depth estimate [8]. Finally, the fourth method adapted optical flow techniques to estimate the disparity on row or columns of the light field [9].

Figures E.1 to E.32 show the results for the 4 methods and 8 different light fields. For each method, the depth or disparity was estimated for the centre SAI of the light field decoded with the toolbox of Dansereau et al. [6] without (*Da*) and with denoising (*DaN*), our demultiplexing (*De*), and our full pipeline without (*Re*) and with denoising (*ReN*). Note that all results were colour coded so that close objects appear in white, while far objects appear black.

Since no ground truth is available for the depth or disparity maps, no objective evaluation could be conducted. For each method, slight variations can be observed between the depth or disparity maps corresponding to the different steps, but no step seems to clearly deter or improve the performances. Note that this is also true after the denoising step, even though denoising is sometimes not recommended before such application. While in general denoising may smooth images, the LFBM5D algorithm chosen in this paper can preserve edges, which are useful features for the depth or disparity estimation. Thus the proposed pipeline does not seem to strongly impact the performances of depth or disparity map estimation.

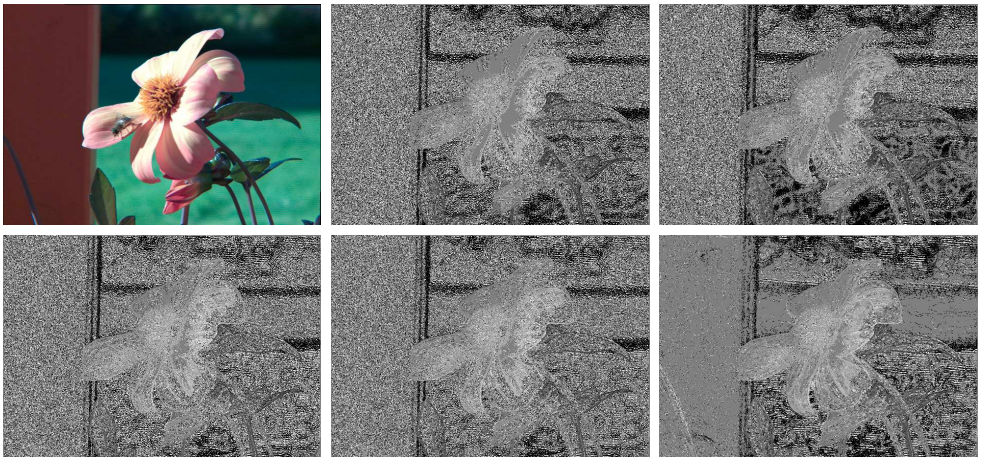


Figure E.1: Depth map estimated with [6] on *bee_1*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

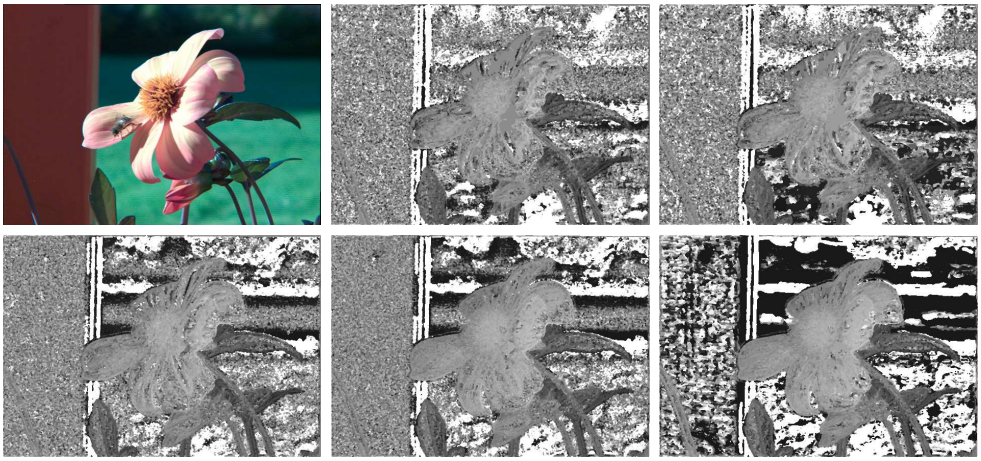


Figure E.2: Depth map estimated with [7] on *bee_1*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

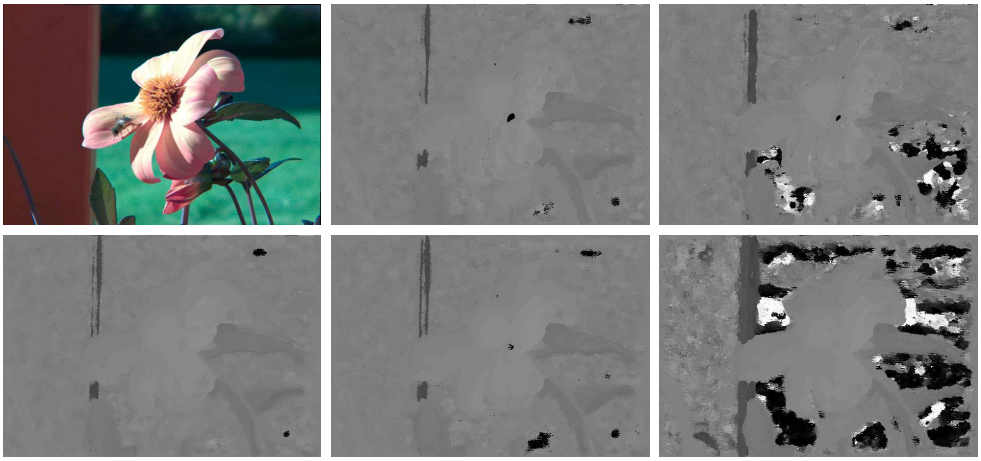


Figure E.3: Depth map estimated with [8] on *bee_1*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

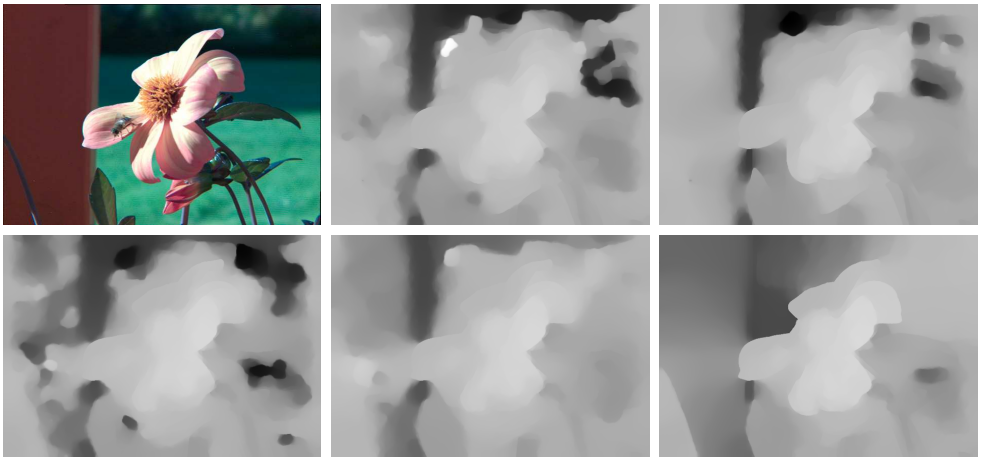


Figure E.4: Disparity map estimated with [9] on *bee_1*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

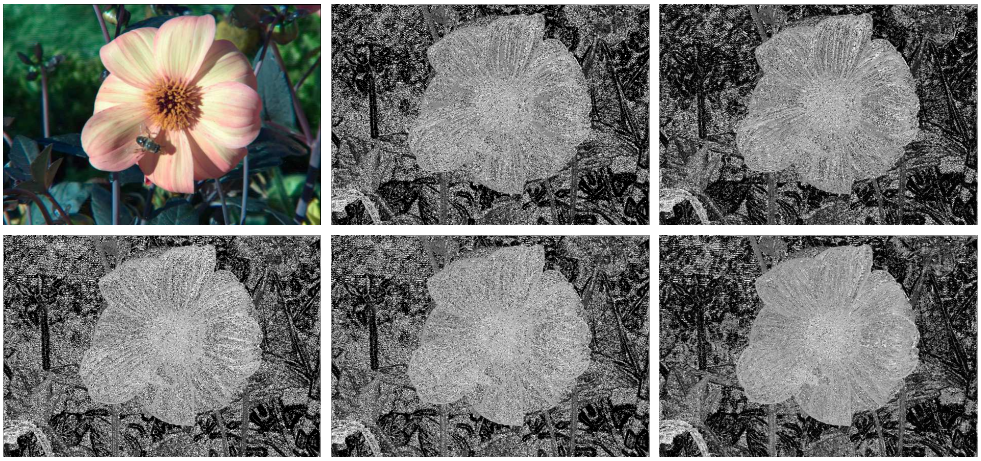


Figure E.5: Depth map estimated with [6] on *bee_2*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

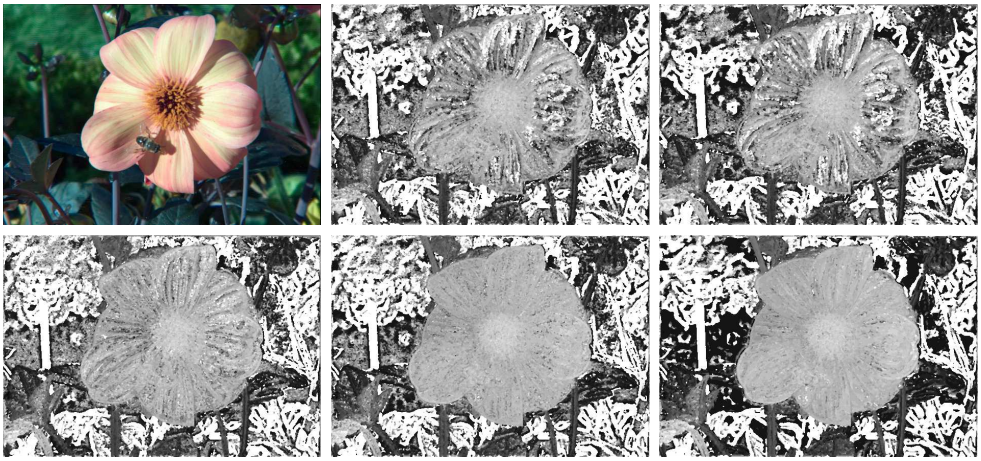


Figure E.6: Depth map estimated with [7] on *bee_2*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

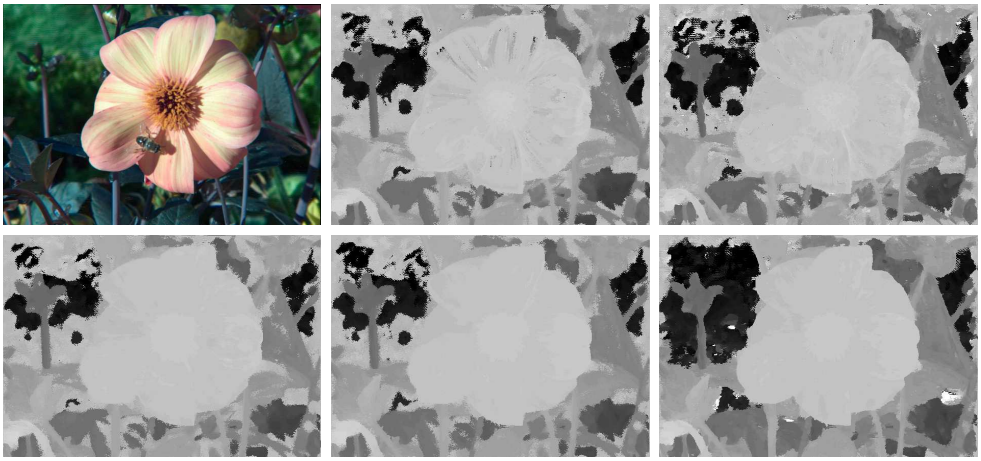


Figure E.7: Depth map estimated with [8] on *bee_2*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

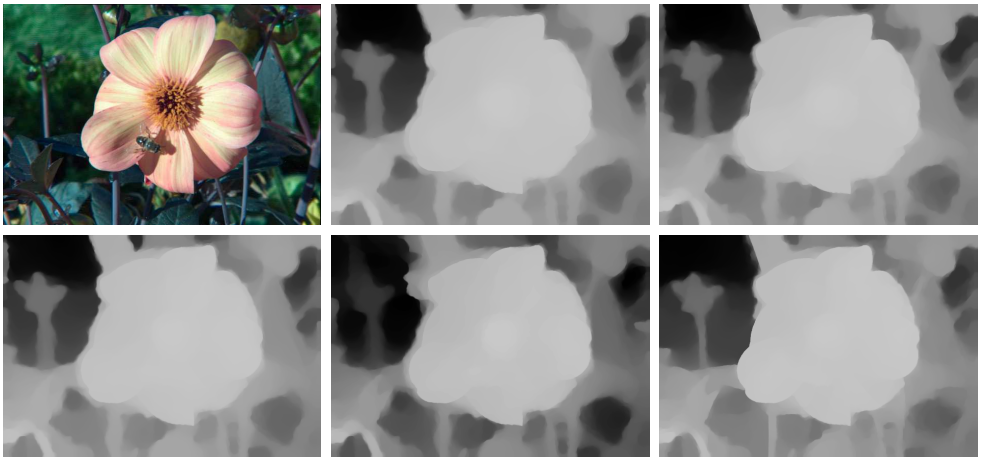


Figure E.8: Disparity map estimated with [9] on *bee_2*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

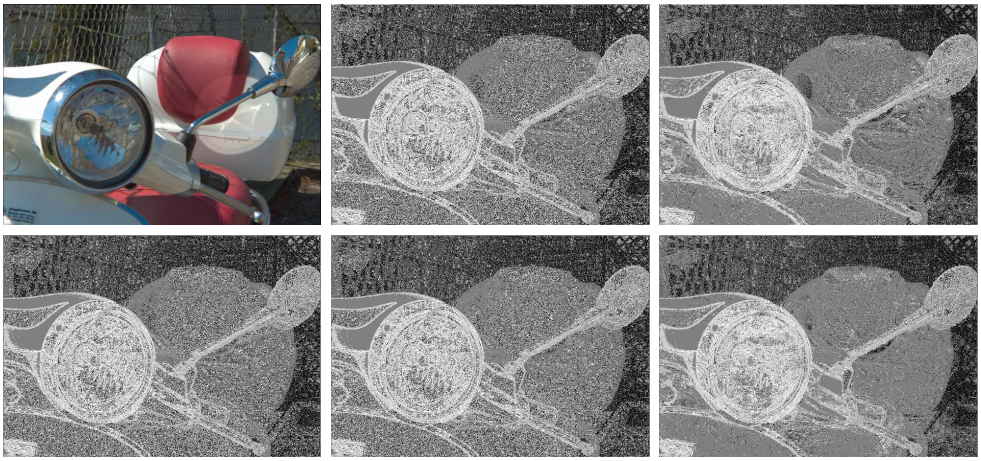


Figure E.9: Depth map estimated with [6] on *vespa*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

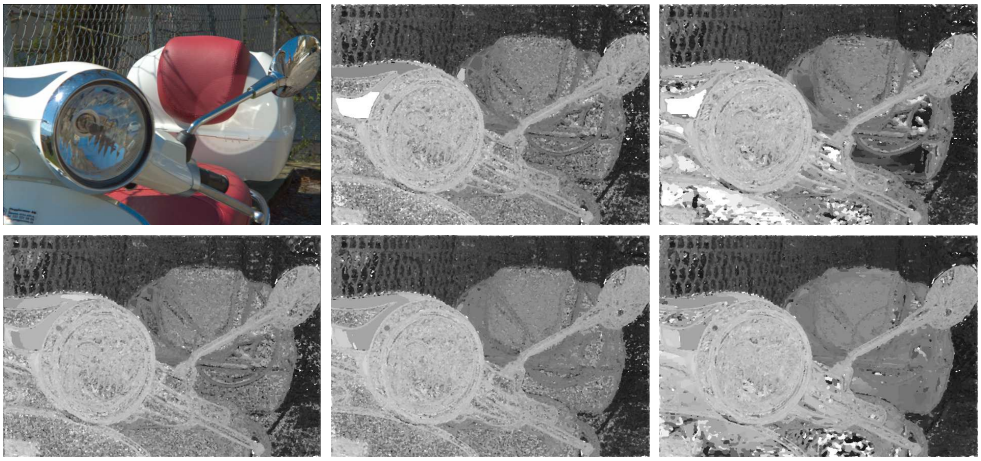


Figure E.10: Depth map estimated with [7] on *vespa*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

Figure E.11: Depth map estimated with [8] on *vespa*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

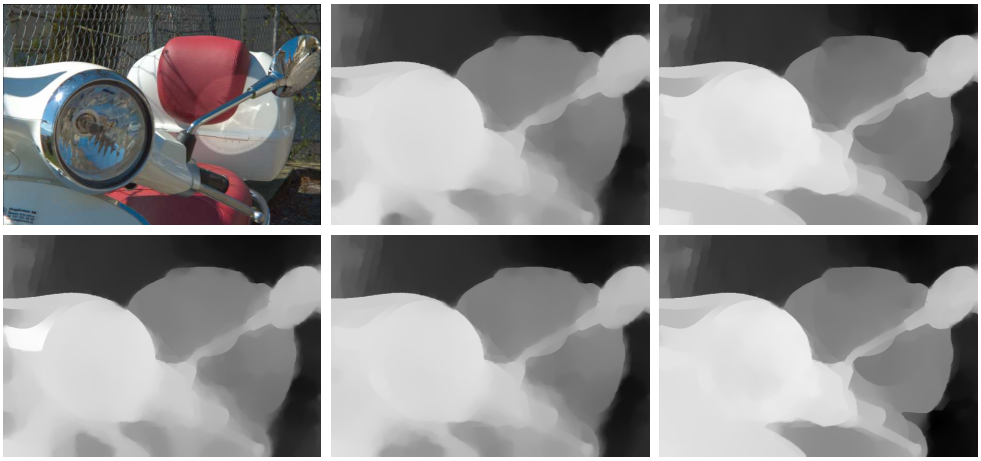


Figure E.12: Disparity map estimated with [9] on *vespa*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

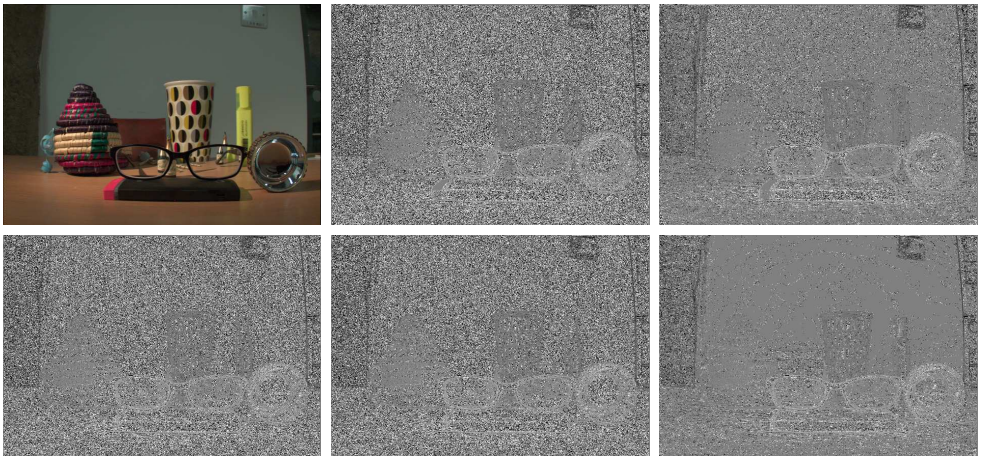


Figure E.13: Depth map estimated with [6] on *glasses1*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

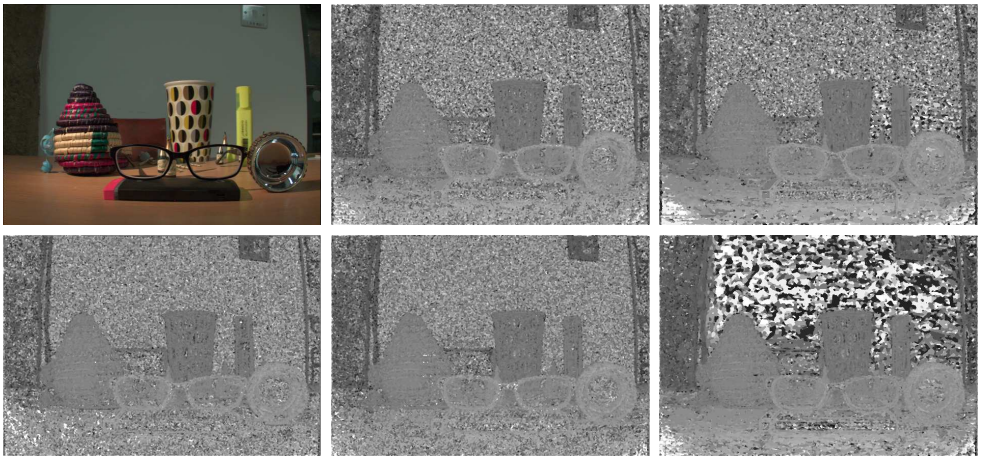


Figure E.14: Depth map estimated with [7] on *glasses1*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

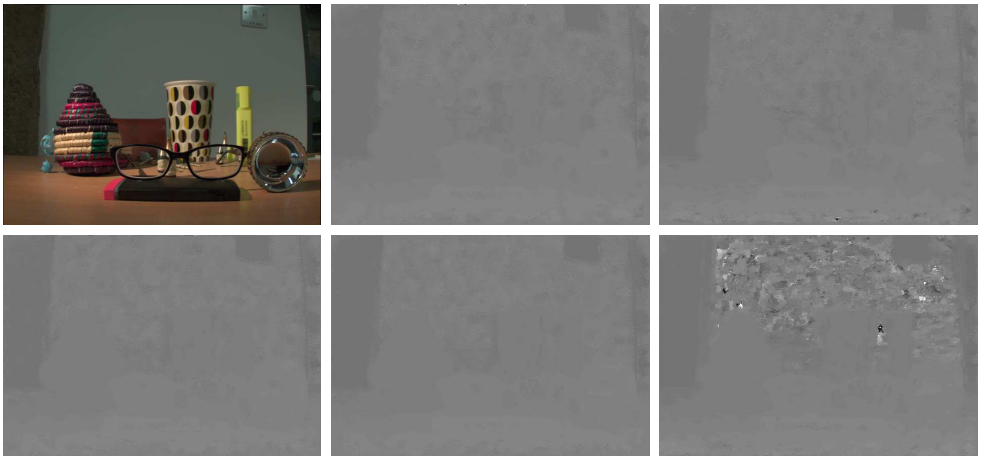


Figure E.15: Depth map estimated with [8] on *glasses1*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

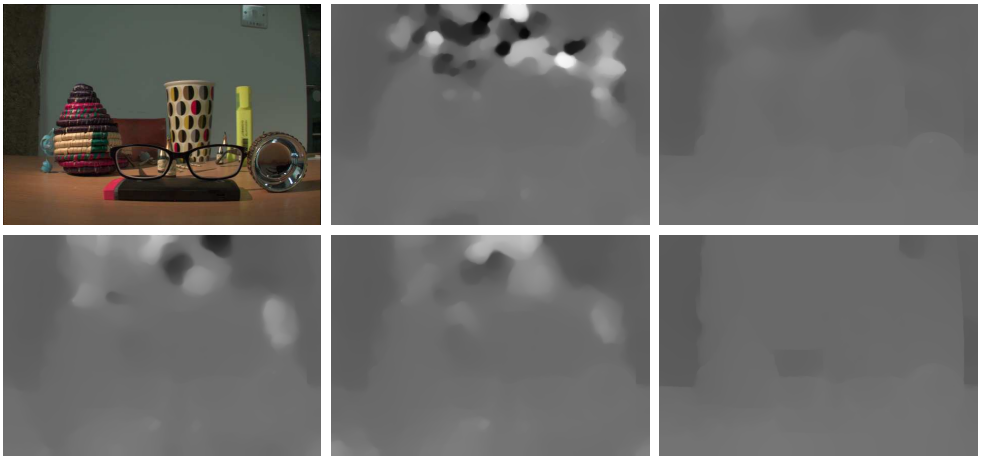


Figure E.16: Disparity map estimated with [9] on *glasses1*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

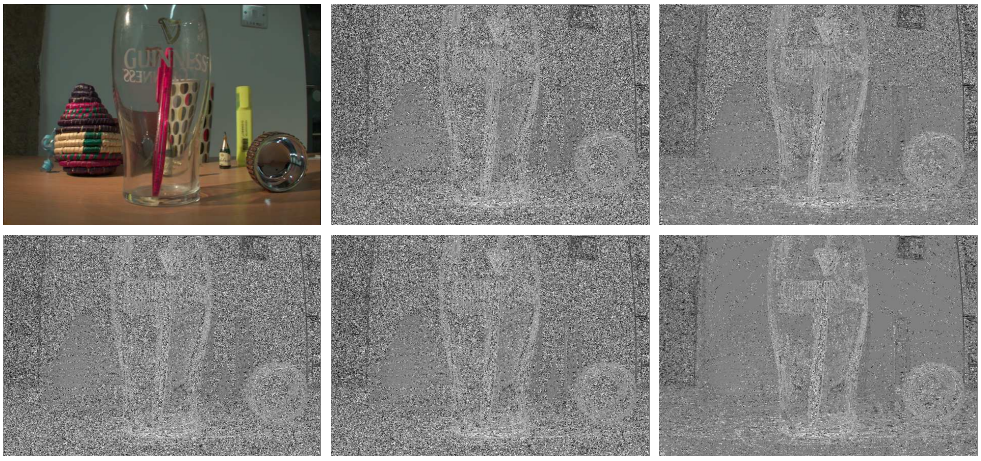


Figure E.17: Depth map estimated with [6] on *guinness*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

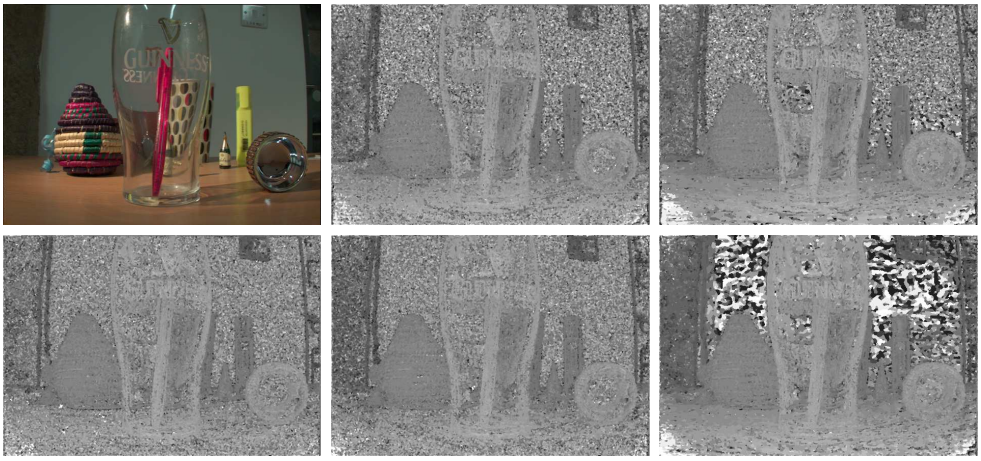


Figure E.18: Depth map estimated with [7] on *guinness*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

Figure E.19: Depth map estimated with [8] on *guinness*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

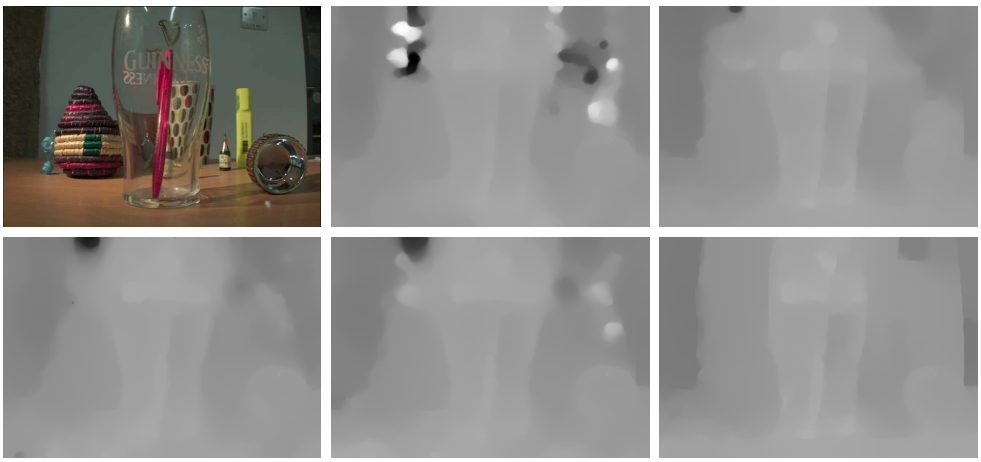


Figure E.20: Disparity map estimated with [9] on *guinness*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

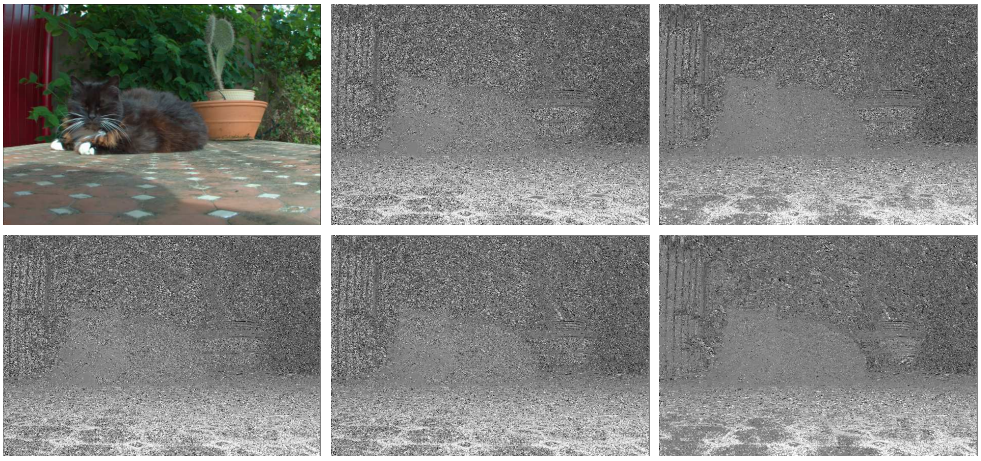


Figure E.21: Depth map estimated with [6] on *odette*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

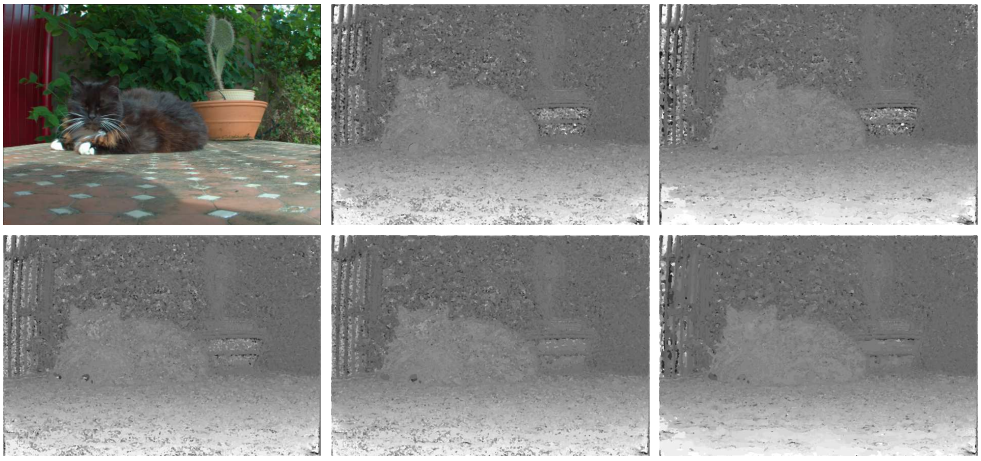


Figure E.22: Depth map estimated with [7] on *odette*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

Figure E.23: Depth map estimated with [8] on *odette*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

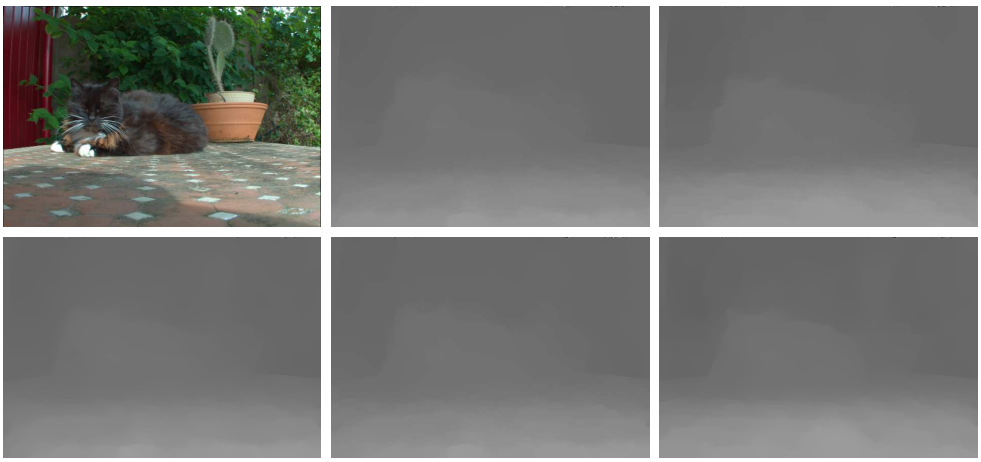


Figure E.24: Disparity map estimated with [9] on *odette*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

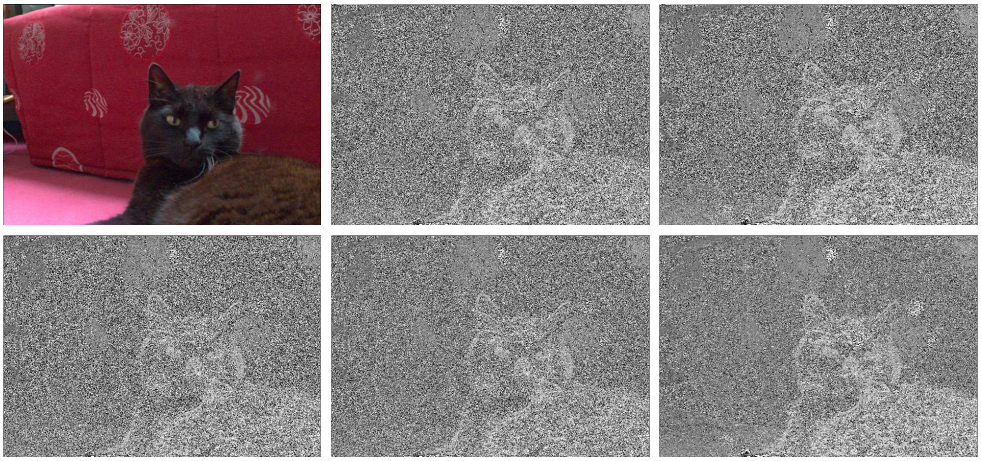


Figure E.25: Depth map estimated with [6] on *raoul*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

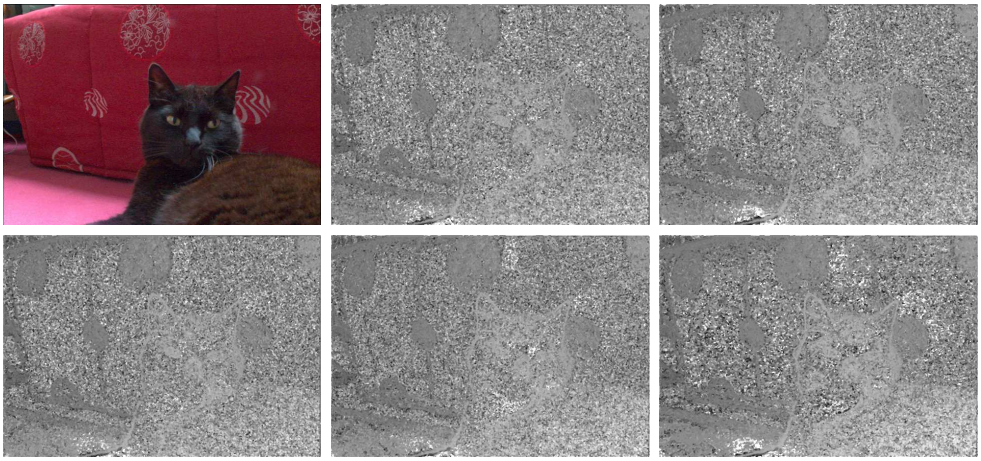


Figure E.26: Depth map estimated with [7] on *raoul*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

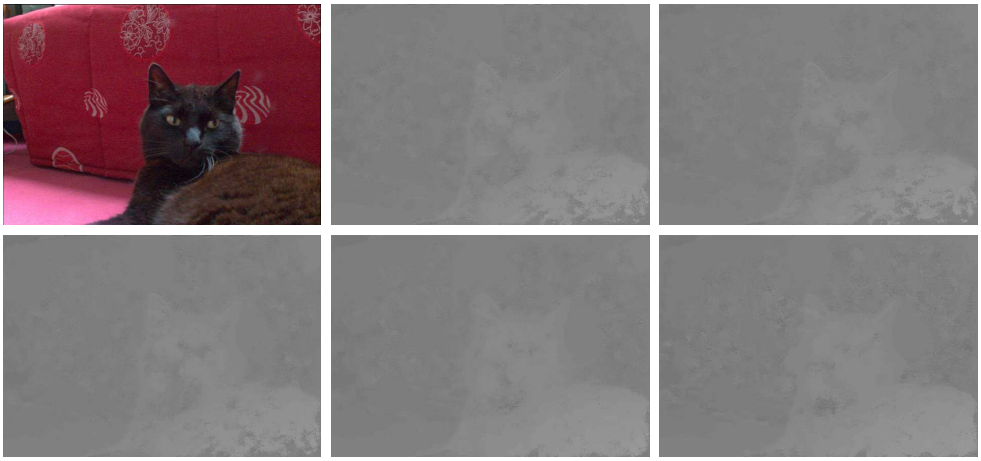


Figure E.27: Depth map estimated with [8] on *raoul*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

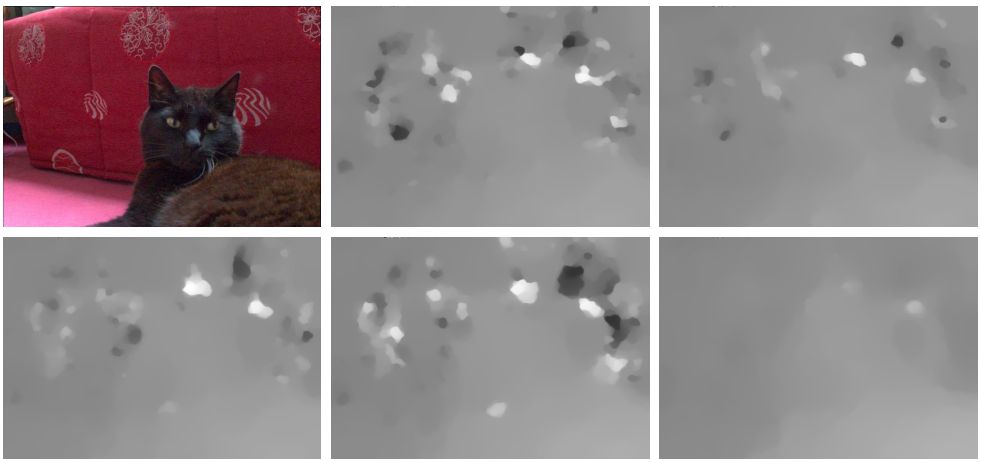


Figure E.28: Disparity map estimated with [9] on *raoul*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

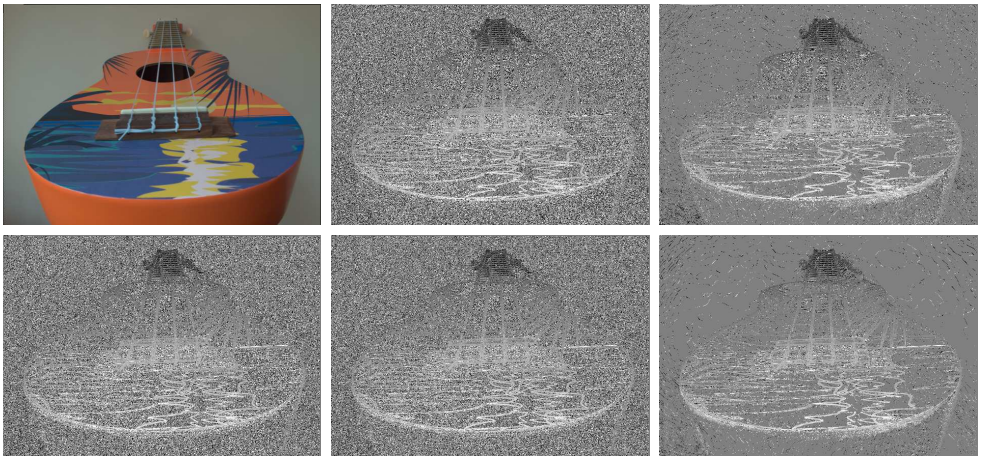


Figure E.29: Depth map estimated with [6] on *ukulele*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

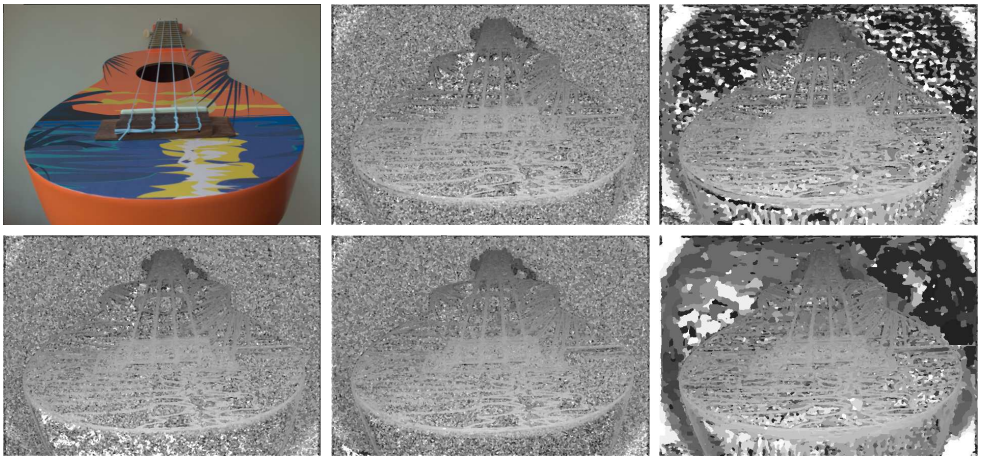


Figure E.30: Depth map estimated with [7] on *ukulele*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

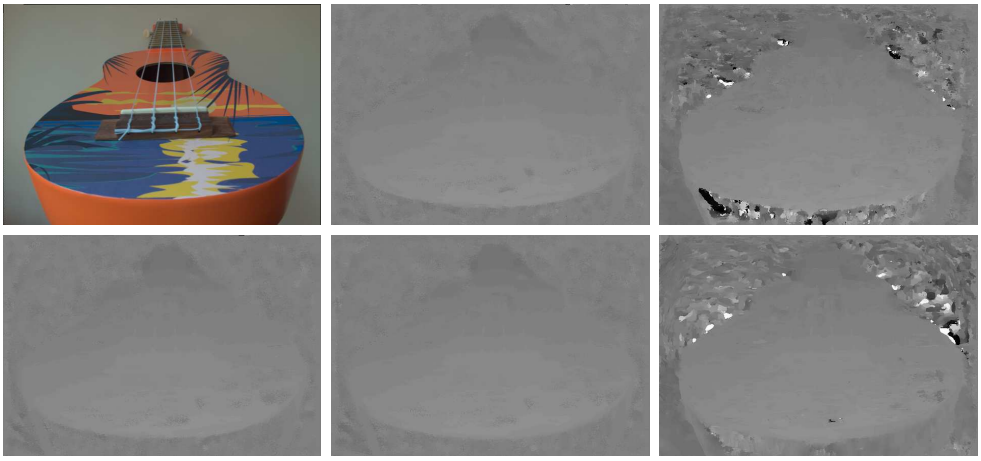


Figure E.31: Depth map estimated with [8] on *ukulele*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

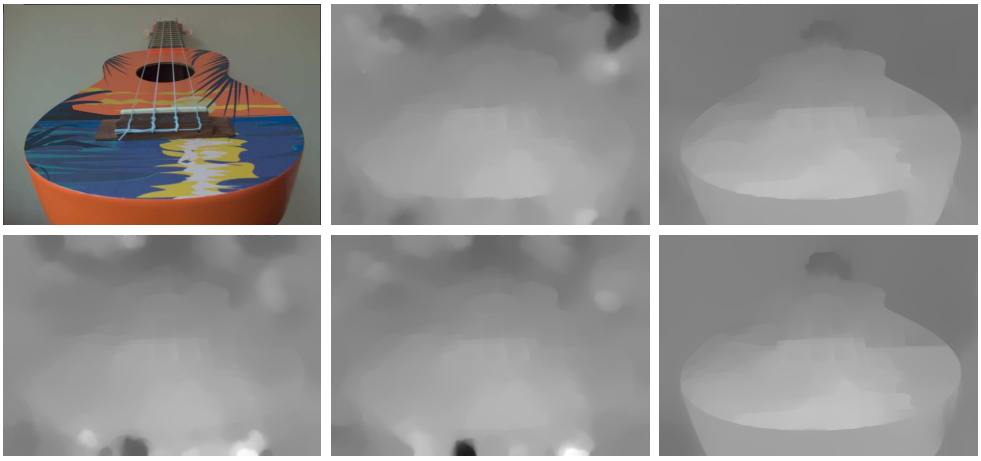


Figure E.32: Disparity map estimated with [9] on *ukulele*. From top to bottom, left to right: centre SAI, *Da*, *DaN*, *De*, *Re*, *ReN*.

# Bibliography


[1] D. Cho, M. Lee, S. Kim, and Y. W. Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Proc. IEEE ICCV*, pp. 3280–3287, Dec. 2013.

[2] S. Xu, Z.-L. Zhou, and N. Devaney, "Multi-view image restoration from plenoptic raw images," in *Proc. ACCV Workshops*, 2014.

[3] M. Seifi, N. Sabater, V. Drazic, and P. Perez, "Disparity-guided demosaicking of light field images," in *Proc. IEEE ICIP*, pp. 5482–5486, Oct. 2014.

[4] C. Hahne, A. Aggoun, V. Velisavljevic, S. Fiebig, and M. Pesch, "Baseline and triangulation geometry in a standard plenoptic camera," *International Journal of Computer Vision*, vol. 126, pp. 21–35, Jan 2018.

[5] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proc. IEEE CVPR*, pp. 1027–1034, 2013.

[6] D. Dansereau and L. Bruton, "Gradient-based depth estimation from 4d light fields," in *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. III–549, May 2004.

[7] T. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 3487–3495, Dec 2015.

[8] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Computer Vision and Image Understanding*, vol. 145, pp. 148 – 159, 2016.

[9] Y. Chen, M. Alain, and A. Smolic, "Fast and accurate optical flow based depth map estimation from light fields," in *Proceedings of the Irish Machine Vision and Image Processing Conference*, 2017.

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

[11] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Transactions on Graphics*, vol. 24, pp. 765–776, July 2005.

[12] "The stanford light field archive." `http://lightfield.stanford.edu/lfs.html`. accessed: 27-12-2021.

[13] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light Field Photography with a Hand-Held Plenoptic Camera," tech. rep., Stanford University CSTR, Apr. 2005.

[14] J. Trottnow, S. Spielmann, T. Herfet, T. Lange, K. Chelli, M. Solony, P. Smrz, P. Zemcik, W. Aenchbacher, M. Grogan, M. Alain, A. Smolic, T. Canham, O. Vu-Thanh, J. Vázquez-Corral, and M. Bertalmío, "The potential of light fields in media productions," in *SIGGRAPH Asia 2019 Technical Briefs*, SA '19, (New York, NY, USA), p. 71–74, Association for Computing Machinery, 2019.

[15] A. Jarabo, B. Masia, A. Bousseau, F. Pellacini, and D. Gutierrez, "How do people edit light fields?," *ACM TOG*, vol. 33, no. 4, 2014.

[16] Y. Aksoy, T. O. Aydın, M. Pollefeys, and A. Smolić, "Interactive high-quality green-screen keying via color unmixing," *ACM Trans. Graph.*, vol. 35, no. 5, pp. 152:1–152:12, 2016.

[17] Y. Aksoy, T. O. Aydin, A. Smolić, and M. Pollefeys, "Unmixing-based soft color segmentation for image manipulation," *ACM Trans. Graph.*, vol. 36, Mar. 2017.

[18] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for

view synthesis," in *The European Conference on Computer Vision (ECCV)*, 2020.

[19] P. Matysiak, M. Grogan, M. L. Pendu, M. Alain, and A. Smolic, "A pipeline for lenslet light field quality enhancement," in *Proc. IEEE ICIP*, pp. 639–643, Oct. 2018.

[20] P. Matysiak, M. Grogan, M. Le Pendu, M. Alain, E. Zerman, and A. Smolic, "High quality light field extraction and post-processing for raw plenoptic data," *IEEE Transactions on Image Processing*, pp. 1–1, 2020.

[21] P. Matysiak, M. Grogan, W. Aenchbacher, and A. Smolic, "Soft colour segmentation on light fields," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2621–2625, 2020.

[22] M. S. Zubairy, "A very brief history of light," in *Optics in Our Time* (M. D. Al-Amri, M. El-Gomati, and M. S. Zubairy, eds.), pp. 3–24, Springer International Publishing, 2016.

[23] M. Faraday, *Experimental Researches in Electricity Vol III*. Taylor & Francis (London), May 1846.

[24] L. Guilmette, "The history of maxwell's equations," in *Writing Across the Curriculum. 3.*, 2012.

[25] P. Bouguer, *Essai d'optique, sur la gradation de la lumière*. Claude Jombert (Paris), 1729.

[26] I. H. Lambert, *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. ETH-Bibliothek Zürich, Rar 1355, 1760.

[27] A. Gershun, "The light field," *Journal of Mathematics and Physics*, vol. 18, no. 1-4, pp. 51–151, 1939.

[28] S. Chandrasekhar, *Radiative Transfer*. Dover Publications (New York), 1950.

[29] G. B. Rybicki, "Radiative transfer," *Journal of Astrophysics and Astronomy*, vol. 17, pp. 95–112, 1996.

[30] M. Levoy, “Light fields and computational imaging,” *Computer*, vol. 39, no. 8, pp. 46–55, 2006.

[31] J. T. Kajiya, “The rendering equation,” *SIGGRAPH Comput. Graph.*, vol. 20, p. 143–150, aug 1986.

[32] E. H. Adelson and J. R. Bergen, “The plenoptic function and the elements of early vision,” in *Computational Models of Visual Processing*, pp. 3–20, MIT Press, 1991.

[33] I. Ihrke, J. Restrepo, and L. Mignard-Debise, “Principles of Light Field Imaging: Briefly revisiting 25 years of research,” *IEEE Signal Processing Magazine*, vol. 33, pp. 59–69, Sept. 2016.

[34] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, “The lumigraph,” *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[35] S. E. Chen, “Quicktime vr: An image-based approach to virtual environment navigation,” in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’95, (New York, NY, USA), p. 29–38, Association for Computing Machinery, 1995.

[36] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, “Plenoptic sampling,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’00, (USA), p. 307–318, ACM Press/Addison-Wesley Publishing Co., 2000.

[37] A. Isaksen, L. McMillan, and S. J. Gortler, “Dynamically reparameterized light fields,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’00, (USA), p. 297–306, ACM Press/Addison-Wesley Publishing Co., 2000.

[38] F. Durand, N. Holzschuch, C. Soler, E. Chan, and F. X. Sillion, “A frequency analysis of light transport,” *ACM Trans. Graph.*, vol. 24, p. 1115–1126, jul 2005.

[39] R. Ng, “Fourier slice photography,” *ACM Trans. Graph.*, vol. 24, p. 735–744, jul 2005.

[40] M. Zwicker, W. Matusik, F. Durand, H. Pfister, and C. Forlines, “Antialiasing for automultiscopic 3d displays,” in *ACM SIGGRAPH 2006 Sketches*, SIGGRAPH ’06, (New York, NY, USA), p. 107–es, Association for Computing Machinery, 2006.

[41] R. Ramamoorthi, D. Mahajan, and P. Belhumeur, “A first-order analysis of lighting, shading, and shadows,” *ACM Trans. Graph.*, vol. 26, p. 2–es, jan 2007.

[42] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, “The digital michelangelo project: 3d scanning of large statues,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’00, (USA), p. 131–144, ACM Press/Addison-Wesley Publishing Co., 2000.

[43] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, “Surface light fields for 3d photography,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’00, (USA), p. 287–296, ACM Press/Addison-Wesley Publishing Co., 2000.

[44] H.-Y. Shum and L.-W. He, “Rendering with concentric mosaics,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’99, (USA), p. 299–306, ACM Press/Addison-Wesley Publishing Co., 1999.

[45] T. Herfet, T. Lange, and H. P. Hariharan, “Enabling multiview- and light field-video for veridical visual experiences,” in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 1705–1709, 2018.

[46] G. Lippmann, “”Épreuves réversibles. photographies intégrales”,” in *Comptes Rendus de l’Académie des Sciences*, vol. 146, pp. 446—451, 1908.

[47] A. Lumsdaine and T. Georgiev, “The focused plenoptic camera,” in *Proc. IEEE ICCP*, pp. 1–8, 2009.

[48] “K|lens one - light field lens.” https://www.k-lens-one.com/en/home. last accessed: 27-12-2021.

[49] F. correspondent of the Scientific American”, “”integral photography - a new discovery by professor lippmann”,” in *Scientific American*, vol. 105, p. 164, 1911.

[50] K. Timby, *3D and Animated Lenticular Photography: Between Utopia and Entertainment*. De Gruyter, 2015.

[51] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, “Light field microscopy,” in *ACM SIGGRAPH 2006 Papers*, SIGGRAPH ’06, (New York, NY, USA), p. 924–934, Association for Computing Machinery, 2006.

[52] F. Huang, K. Chen, and G. Wetzstein, “The Light Field Stereoscope: Immersive Computer Graphics via Factored Near-Eye Light Field Displays with Focus Cues,” *ACM Trans. Graph. (SIGGRAPH)*, no. 4, 2015.

[53] D. Brewster, *The Stereoscope: Its History, Theory, and Construction, with Its Application to the Fine and Useful Arts and to Education*. J. Murray (London), 1856.

[54] M. Ziegler, A. Engelhardt, S. Müller, J. Keinert, F. Zilly, S. Foessel, and K. Schmid, “Multi-camera system for depth based visual effects and compositing,” in *Proceedings of the 12th European Conference on Visual Media Production*, CVMP ’15, (New York, NY, USA), Association for Computing Machinery, 2015.

[55] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, “Unstructured lumigraph rendering,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’01, (New York, NY, USA), p. 425–432, Association for Computing Machinery, 2001.

[56] A. Davis, M. Levoy, and F. Durand, “Unstructured light fields,” *Comput. Graph. Forum*, vol. 31, p. 305–314, may 2012.

[57] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ra-

mamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Trans. Graph.*, vol. 38, jul 2019.

[58] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," *arXiv preprint arXiv:2107.02791*, 2021.

[59] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," *arXiv preprint arXiv:2111.11215*, 2021.

[60] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 15651–15663, Curran Associates, Inc., 2020.

[61] D. B. Lindell, J. N. Martel, and G. Wetzstein, "Autoint: Automatic integration for fast neural volume rendering," in *Proceedings of the conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[62] D. Rebain, W. Jiang, S. Yazdani, K. Li, K. M. Yi, and A. Tagliasacchi, "Derf: Decomposed radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14153–14161, June 2021.

[63] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. S. Kaplanyan, and M. Steinberger, "DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks," *Computer Graphics Forum*, vol. 40, no. 4, 2021.

[64] S. Lombardi, T. Simon, G. Schwartz, M. Zollhoefer, Y. Sheikh, and J. Saragih, "Mixture of volumetric primitives for efficient neural rendering," 2021.

[65] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," *ICCV*, 2021.

[66] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-

nerf: Neural radiance fields for dynamic scenes," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10313–10322, 2021.

[67] G. Gafni, J. Thies, M. Zollhöfer, and M. Nießner, "Dynamic neural radiance fields for monocular 4d facial avatar reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8649–8658, June 2021.

[68] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt, "Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video," in *IEEE International Conference on Computer Vision (ICCV)*, IEEE, 2021.

[69] A. Raj, M. Zollhoefer, T. Simon, J. Saragih, S. Saito, J. Hays, and S. Lombardi, "Pva: Pixel-aligned volumetric avatars," *arXiv preprint arXiv:2101.02697*, 2021.

[70] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6494–6504, 2021.

[71] W. Xian, J.-B. Huang, J. Kopf, and C. Kim, "Space-time neural irradiance fields for free-viewpoint video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9421–9431, June 2021.

[72] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[73] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic view synthesis from dynamic monocular video," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[74] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. Lensch,

"Nerd: Neural reflectance decomposition from image collections," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[75] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7495–7504, June 2021.

[76] S. Wizadwongsa, P. Phongthawee, J. Yenphraphai, and S. Suwajanakorn, "Nex: Real-time view synthesis with neural basis expansion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[77] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "Nerfactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Trans. Graph.*, vol. 40, dec 2021.

[78] S. Liu, X. Zhang, Z. Zhang, R. Zhang, J.-Y. Zhu, and B. Russell, "Editing conditional radiance fields," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[79] J. Zhang, X. Liu, X. Ye, F. Zhao, Y. Zhang, M. Wu, Y. Zhang, L. Xu, and J. Yu, "Editable free-viewpoint video using a layered neural representation," *ACM Trans. Graph.*, vol. 40, jul 2021.

[80] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF: Inverting neural radiance fields for pose estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[81] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[82] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park, "Self-calibrating neural radiance fields," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[83] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[84] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, "Neural scene graphs for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2856–2865, June 2021.

[85] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui, "Learning object-compositional neural radiance field for editable scene rendering," in *International Conference on Computer Vision (ICCV)*, October 2021.

[86] H. Baatz, J. Granskog, M. Papas, F. Rousselle, and J. Novák, "NeRF-Tex: Neural Reflectance Field Textures," in *Eurographics Symposium on Rendering - DL-only Track* (A. Bousseau and M. McGuire, eds.), The Eurographics Association, 2021.

[87] C. Xie, K. Park, R. Martin-Brualla, and M. Brown, "Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling," in *International Conference on 3D Vision (3DV)*, 2021.

[88] "JPEG Pleno call for proposals on light field coding," Oct. 2016.

[89] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 926–954, 2017.

[90] P. David, M. L. Pendu, and C. Guillemot, "White lenslet image guided demosaicing for plenoptic cameras," in *Proceedings of the IEEE Int. Workshop on Multimedia Signal Processing*, pp. 1–6, Oct. 2017.

[91] Z. Yu, J. Yu, A. Lumsdaine, and T. Georgiev, "An analysis of color demosaicing in plenoptic cameras," in *Proc. IEEE CVPR*, pp. 901–908, Jun. 2012.

[92] X. Huang and O. Cossairt, "Dictionary learning based color demosaicing for plenoptic cameras," in *Proc. IEEE CVPR Workshops*, pp. 455–460, Jun. 2014.

[93] T. Lian and K. Chiang, “Demosaicing and denoising on simulated light field images,” 2016.

[94] M. Grogan and R. Dahyot, “L2 divergence for robust colour transfer,” *Computer Vision and Image Understanding*, 2019.

[95] M. Oliveira, A. Sappa, and V. Santos, “A probabilistic approach for color correction in image mosaicking applications,” *IEEE Transactions on Image Processing*, vol. 24, pp. 508–523, Feb 2015.

[96] J. Park, Y. Tai, S. N. Sinha, and I. S. Kweon, “Efficient and robust color consistency for community photo collections,” in *Proc. IEEE CVPR*, pp. 430–438, June 2016.

[97] M. Xia, J. Y. Renping, X. M. Zhang, and J. Xiao, “Color consistency correction based on remapping optimization for image stitching,” in *Proc. IEEE ICCV Workshops*, pp. 2977–2984, Oct 2017.

[98] Y. Hwang, J.-Y. Lee, I. S. Kweon, and S. J. Kim, “Probabilistic moving least squares with spatial constraints for nonlinear color transfer between images,” *Computer Vision and Image Understanding*, 2019.

[99] J. Fu, Y. Wu, X. Mou, W. Ji, and P. Wang, “Fpga-based implementation of estimating saturated pixel values in RAW image,” in *Digital Photography and Mobile Imaging XII*, pp. 1–6, 2016.

[100] M. Assefa, T. Pouli, J. Kervec, and M. Larabi, “Correction of overexposure using color channel correlations,” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1078–1082, Dec. 2014.

[101] E. Elboher and M. Werman, “Recovering color and details of clipped image regions,” in *Proc. CGVCVIP*, 2010.

[102] S. Z. Masood, J. Z., and M. F. Tappen, “Automatic correction of saturated regions in photographs using cross-channel correlation,” *Comput. Graph. Forum*, vol. 28, pp. 1861–1869, Oct. 2009.

[103] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[104] L. Shao, R. Yan, X. Li, and Y. Liu, “From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms,” *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1001–1013, 2014.

[105] P. Jain and V. Tyagi, “A survey of edge-preserving image denoising methods,” *Information Systems Frontiers*, vol. 18, pp. 159–170, 2016.

[106] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, “Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms,” *IEEE TIP*, vol. 21, no. 9, pp. 3952–3966, 2012.

[107] Z. Li, H. Baker, and R. Bajcsy, “Joint image denoising using light-field data,” in *Proc. IEEE ICME Workshops*, 2013.

[108] A. Sepas-Moghaddam, P. L. Correia, and F. Pereira, “Light field denoising: exploiting the redundancy of an epipolar sequence representation,” in *Proceedings of the 3DTV Conference*, pp. 1–4, Jul 2016.

[109] J. Chen, J. Hou, and L. Chau, “Light field denoising via anisotropic parallax analysis in a cnn framework,” *IEEE Signal Processing Letters*, vol. 25, pp. 1403–1407, Sep. 2018.

[110] Y. Liu, N. Qi, Z. Cheng, D. Liu, Q. Ling, and Z. Xiong, “Tensor-based light field denoising by integrating super-resolution,” in *Proc. IEEE ICIP*, pp. 3209–3213, Oct 2018.

[111] H. S. Malvar, L.-W. He, and R. Cutler, “High-quality linear interpolation for demosaicing of bayer-patterned color images,” in *Proc. IEEE ICASSP*, 2004.

[112] M. Alain and A. Smolic, “Light field denoising by sparse 5D transform domain collaborative filtering,” in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pp. 1–6, Oct. 2017.

[113] A. R. Robertson, “The CIE 1976 color-difference formulae,” *Color Research & Application*, vol. 2, no. 1, pp. 7–11, 1977.

[114] F. Pitié, A. C. Kokaram, and R. Dahyot, “Automated colour grading using colour distribution transfer,” *Computer Vision and Image Understanding*, vol. 107, pp. 123–137, July 2007.

[115] S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol, "Regularized discrete optimal transport," in *Scale Space and Variational Methods in Computer Vision*, pp. 428–439, 2013.

[116] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon wasserstein barycenters of measures," *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 22–45, Jan 2015.

[117] Y. Hwang, J.-Y. Lee, I. S. Kweon, and S. J. Kim, "Color transfer using probabilistic moving least squares," in *Proc. IEEE CVPR*, pp. 3342–3349, June 2014.

[118] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *Proc. IEEE CVPR*, pp. 5704–5712, 2016.

[119] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," in *ACM Transactions on Graphics (TOG)*, vol. 28, p. 24, 2009.

[120] S. H. Park, H. S. Kim, S. Lansel, M. Parmar, and B. A. Wandell, "A case for denoising before demosaicking color filter array data," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pp. 860–864, Nov. 2009.

[121] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM ToG*, vol. 35, pp. 191:1–191:12, Nov. 2016.

[122] L. Condat and S. Mosaddegh, "Joint demosaicking and denoising by total variation minimization," in *IEEE International Conference on Image Processing*, pp. 2781–2784, Sep. 2012.

[123] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *Proceedings of the International Conference on Quality of Multimedia Experience*, 2016.

[124] M. L. Pendu, X. Jiang, and C. Guillemot, "Light field inpainting propagation via low rank matrix completion," *IEEE Transactions on Image Processing*, vol. 27, pp. 1981–1993, April 2018.

[125] D. G. Dansereau, B. Girod, and G. Wetzstein, “LiFF: Light field features in scale and depth,” in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, June 2019.

[126] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[127] X. Zhang and B. A. Wandell, “A spatial extension of CIELAB for digital color-image reproduction,” *Journal of the Society for Information Display*, vol. 5, no. 1, pp. 61–63, 1997.

[128] G. Chen, F. Zhu, and P. A. Heng, “An efficient statistical method for image noise level estimation,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 477–485, Dec 2015.

[129] ITU-R, “Methodology for the subjective assessment of the quality of television pictures.” ITU-R Recommendation BT.500-13, Jan 2012.

[130] M. Perez-Ortiz and R. K. Mantiuk, “A practical guide and software for analysing pairwise comparison experiments.” arXiv:1712.03686, 2017.

[131] H. Talebi and P. Milanfar, “NIMA: Neural image assessment,” *IEEE Transactions on Image Processing*, vol. 27, pp. 3998–4011, Aug 2018.

[132] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, “Pseudo-sequence-based light field image compression,” in *Proc. IEEE ICME Workshops*, pp. 1–4, July 2016.

[133] G. Bjontegaard, “Calculation of average PSNR differences between RD curves,” *document VCEG-M33, ITU-T VCEG Meeting*, 2001.

[134] M. Alain and A. Smolic, “Light field super-resolution via LFBM5D sparse coding,” in *Proc. IEEE ICIP*, pp. 2501–2505, Oct 2018.

[135] F. Zhang, J. Wang, E. Shechtman, Z. Zhou, J. Shi, and S. Hu, “Plenopatch: Patch-based plenoptic image manipulation,” *IEEE TVCG*, vol. 23, pp. 1561–1573, May 2017.

[136] O. Frigo and C. Guillemot, “Epipolar Plane Diffusion: An Efficient Approach for Light Field Editing,” in *Proc. BMVC*, Sept. 2017.

[137] J. Tan, J.-M. Lien, and Y. Gingold, "Decomposing images into layers via RGB-space geometry," *ACM Transactions on Graphics (TOG)*, vol. 36, pp. 7:1–7:14, Nov. 2016.

[138] J. Tan, J. Echevarria, and Y. Gingold, "Efficient palette-based decomposition and recoloring of images via rgbxy-space geometry," *ACM Transactions on Graphics (TOG)*, vol. 37, pp. 262:1–262:10, Dec. 2018.

[139] Y. Koyama and M. Goto, "Decomposing images into layers with advanced color blending," *Computer Graphics Forum*, vol. 37, no. 7, pp. 397–407, 2018.

[140] Y. Wang, Y. Liu, and K. Xu, "An improved geometric approach for palette-based image decomposition and recoloring," *Computer Graphics Forum*, vol. 38, no. 7, pp. 11–22, 2019.

[141] T. Jeong, M. Yang, and H. J. Shin, "Succinct palette and color model generation and manipulation using hierarchical representation," *Computer Graphics Forum*, vol. 38, no. 7, pp. 1–10, 2019.

[142] H. Mihara, T. Funatomi, K. Tanaka, H. Kubo, Y. Mukaigawa, and H. Nagahara, "4d light field segmentation with spatial and angular consistencies," in *2016 IEEE International Conference on Computational Photography (ICCP)*, pp. 1–8, May 2016.

[143] M. Hog, N. Sabater, and C. Guillemot, "Light field segmentation using a ray-based graph structure," in *European Conference on Computer Vision*, pp. 35–50, Springer, 2016.

[144] H. Zhu, Q. Zhang, Q. Wang, and H. Li, "4d light field superpixel and segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 85–99, 2020.

[145] N. Khan, Q. Zhang, L. Kasser, H. Stone, M. H. Kimm, and J. Tompkin, "View-consistent 4d lightfield superpixel segmentation," *International Conference on Computer Vision*, 2019.

[146] F. Zhang, J. Wang, E. Shechtman, Z. Zhou, J. Shi, and S. Hu, "Plenopatch: Patch-based plenoptic image manipulation," *IEEE*

*Transactions on Visualization and Computer Graphics*, vol. 23, pp. 1561–1573, May 2017.

[147] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields," in *Asian Conference on Computer Vision*, Springer, 2016.

[148] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, "Light field reconstruction using sparsity in the continuous fourier domain," *ACM Trans. Graph.*, vol. 34, Dec. 2015.

[149] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Light field reconstruction using shearlet transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 133–147, 2018.

[150] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, vol. 35, no. 6, 2016.

[151] Y. Wang, F. Liu, Z. Wang, G. Hou, Z. Sun, and T. Tan, "End-to-end view synthesis for light field imaging with pseudo 4dcnn," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 340–355, Springer International Publishing, 2018.

[152] H. W. F. Yeung, J. Hou, J. Chen, Y. Y. Chung, and X. Chen, "Fast light field reconstruction with deep coarse-to-fine modeling of spatial-angular clues," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 138–154, Springer International Publishing, 2018.

[153] Y. Chen, M. Alain, and A. Smolic, "Self-supervised light field view synthesis using cycle consistency," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2020.

[154] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *ACM Trans. Graph.*, vol. 37, jul 2018.

[155] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck,

N. Snavely, and R. Tucker, "Deepview: View synthesis with learned gradient descent," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2362–2371, 2019.

[156] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely, "Pushing the boundaries of view extrapolation with multiplane images," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 175–184, 2019.

[157] Z. Yu, X. Guo, H. Lin, A. Lumsdaine, and J. Yu, "Line assisted light field triangulation and stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[158] I. Tosic and K. Berkner, "Light field scale-depth space transform for dense depth estimation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 441–448, 2014.

[159] Y. Luo, W. Zhou, J. Fang, L. Liang, H. Zhang, and G. Dai, "Epi-patch based convolutional neural network for depth estimation on 4d light field," in *Neural Information Processing* (D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, eds.), (Cham), pp. 642–652, Springer International Publishing, 2017.

[160] M. Feng, Y. Wang, J. Liu, L. Zhang, H. F. M. Zaki, and A. Mian, "Benchmark data set and method for depth estimation from light field images," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3586–3598, 2018.

[161] X. Jiang, J. Shi, and C. Guillemot, "A learning based depth estimation framework for 4d densely and sparsely sampled light fields," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2257–2261, 2019.

[162] C. Shin, H.-G. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim, "Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4748–4757, 2018.

[163] N. Khan, M. H. Kim, and J. Tompkin, "Edge-aware bidirectional diffusion for dense depth estimation from light fields," 2021.

[164] Y.-J. Tsai, Y.-L. Liu, M. Ouhyoung, and Y.-Y. Chuang, "Attention-based view selection networks for light-field disparity estimation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12095–12103, Apr. 2020.

[165] J. Chen, S. Zhang, and Y. Lin, "Attention-based multi-level fusion network for light field depth estimation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1009–1017, May 2021.

[166] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo," in *ICCV*, 2021.

[167] W. Aenchbacher, P. Matysiak, and A. Smolic, "A fully-parameterized object-side light field dataset and theory for using entrance and exit pupils as natural light field reference planes for an unfocused plenoptic camera," 2022.

[168] N. Sabater, G. Boisson, B. Vandame, P. Kerbiriou, F. Babon, M. Hog, T. Langlois, R. Gendrot, O. Bureller, A. Schubert, and V. Allie, "Dataset and pipeline for multi-view light-field video," in *CVPR Workshops*, 2017.