

# Four-Dimensional Sparse Filters for Near Real-Time Light Field Processing

I. W. A. S. U. Premaratne

(168050V)

Thesis submitted in partial fulfillment of the requirements for the degree  
Master of Philosophy

Department of Electronic and Telecommunication Engineering

University of Moratuwa  
Sri Lanka

May 2019

## Declaration

I declare that this is my own work, and this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning, and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part, in print, electronic, or any other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

The candidate, whose signature appears above, carried out research for the MPhil dissertation under my supervision.

Signature:

Date:

Dr. C. U. S. Edussooriya

## Abstract

Light is a fundamental form of conveying information. Sensing of light through conventional cameras leads to images and videos. In contrast to conventional images and videos, which capture only the directional variation of the intensity of light rays emanating from a scene, *light fields* capture the spatial variation as well. This richness of information has been exploited to accomplish novel tasks that are not possible with conventional images and videos, such as post-capture digital refocusing and depth filtering.

As a result of the massive data volume captured by a light field, the light field processing algorithms require higher memory and computational requirement. This is a major drawback for employing light fields in real-time applications. Hence, there is a need for investigating novel low-complexity light field processing algorithms that can be implemented in real-time applications. In this study, we address this critical research problem using multidimensional linear filter theory to develop novel low-complexity and sparse filters for light field processing. To this end, the work presented in this thesis focus on two major scenarios; light field denoising and volumetric refocusing. First, we present a novel low-complexity light field denoising algorithm, utilizing the sparsity of the region of support of a light field in the frequency domain. It turns out that the proposed filter runs in near real-time, compared to the previously reported light field denoising methods which take minutes. Next, a 4-D sparse filter for volumetric refocusing is presented. The proposed sparse filter provides 72% reduction of computational complexity compared to a non-sparse filter, with negligible distortion in fidelity.

***Index terms***— light field, Denoising, volumetric refocusing, real-time, sparse filters, low-complexity

## Acknowledgements

I would like to extend my heartfelt gratitude to my supervisor, Dr. Chamira Edussooriya, Department of Electronic and Telecommunication Engineering, University of Moratuwa, for his academic advice, continuous guidance and the tremendous support given to me throughout my research study.

Furthermore, I would also like to thank my progress review panel members Dr. Ranga Rodrigo and Dr. Chamitha de Alwis (University of Sri Jayawardenepura) for their comments and suggestions to improve my research work. Also, I am grateful to the final examination committee, Prof. Sandun Fernando, Dr. Donald Dansereau (University of Sydney) and Dr. Prathapasinghe Dharmawansha, for conducting the examination and for their constructive remarks on the thesis.

Moreover, I would like to acknowledge the financial support received from the Senate Research Council of University of Moratuwa, under the grant number SRC/LT/2016/10.

Finally, I would like to express my sincere gratitude to my family, colleagues and friends, for their love, kind support and encouragement.



# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Abbreviations</b>	<b>xi</b>
<b>List of Algorithms</b>	<b>xii</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Contributions of the Thesis . . . . .	3
1.2 Outline of the Thesis . . . . .	3
<b>2 REVIEW ON LIGHT FIELD IMAGING</b>	<b>5</b>
2.1 Plenoptic Function . . . . .	5
2.2 Light Fields and Applications . . . . .	6
2.3 Light Field Acquisition . . . . .	8
2.4 Noise in Light Fields . . . . .	9
2.5 Denoising Light Fields . . . . .	10
2.5.1 Complexity of Light Field Denoising Methods . . . . .	11
2.6 Light Field Refocusing . . . . .	12
2.6.1 Complexity of Volumetric Refocusing of Light Fields . . . . .	13
<b>3 A LOW-COMPLEXITY DENOISING ALGORITHM FOR LIGHT FIELDS</b>	<b>14</b>

3.1	Introduction . . . . .	14
3.2	Light Field Parameterization . . . . .	15
3.2.1	Spherical-Cartesian Parameterization . . . . .	15
3.2.2	Two-Plane Parameterization . . . . .	16
3.3	Review of Light Field Spectrum . . . . .	17
3.3.1	LF Representation of a Lambertian Point Source . . . . .	17
3.3.2	ROS of the Spectrum . . . . .	18
3.3.3	Dimensionality Gap in Light Fields . . . . .	19
3.4	Sparsity and Selective Filtering . . . . .	20
3.5	Proposed Low-complexity Algorithm . . . . .	22
3.6	2-D Filter Design . . . . .	24
3.6.1	Parameter Selection . . . . .	27
3.7	Results and Comparison . . . . .	28
3.7.1	Complexity Analysis . . . . .	28
3.7.2	Experimental Results . . . . .	30
<b>4</b>	<b>A 4-D SPARSE FILTER FOR VOLUMETRIC REFOCUSING OF LIGHT FIELDS</b>	<b>36</b>
4.1	Introduction . . . . .	36
4.2	Volumetric Refocusing . . . . .	36
4.3	Proposed Sparse Filter . . . . .	38
4.4	Results and Comparison . . . . .	40
4.4.1	Comparison Between the 4-D Sparse and Nonsparse FIR Hyperfan Filters . . . . .	40
4.4.2	Performance of the 4-D Sparse FIR Hyperfan Filter in Vol- umetric Refocusing . . . . .	43
<b>5</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>48</b>
5.1	Conclusions . . . . .	48
5.2	Future Work . . . . .	49
	<b>References</b>	<b>50</b>
<b>A</b>	<b>Derivation of the Ideal Infinite-Extent Impulse Response <math>g_{uv}^I(n_u, n_v)</math> of Parallelogram Filter</b>	<b>58</b>
<b>B</b>	<b>Derivation of the Ideal Infinite-Extent Impulse Response <math>k_{uv}^I(n_u, n_v)</math> of Circular Filter</b>	<b>59</b>

---

<b>C Derivation of the Bow-tie Shaped Passband</b>	<b>60</b>
<b>D Supplementary Results for the Proposed Low-Complexity De-noising Algorithm</b>	<b>63</b>

## List of Figures

Figure 1.1	7-D plenoptic function . . . . .	2
Figure 2.1	Camera design by Adelson . . . . .	6
Figure 2.2	Camera design by Ng . . . . .	6
Figure 2.3	Lytro Illum light field camera (Source- <a href="https://www.lytro.com/illum/">https://www.lytro.com/illum/</a> ) . . . . .	8
Figure 2.4	Raytrix light field camera (Source- <a href="http://lightfield-forum.com/raytrix/raytrix-r11-3d-lightfield-camera/">http://lightfield-forum.com/raytrix/raytrix-r11-3d-lightfield-camera/</a> ) . . . . .	8
Figure 2.5	Stanford multi-camera array (Source- <a href="http://graphics.stanford.edu/projects/array/">http://graphics.stanford.edu/projects/array/</a> ) . . . . .	9
Figure 3.1	Spherical-cartesian parameterization. . . . .	15
Figure 3.2	The two-plane parameterization (with the locally defined image coordinates $(n_u, n_v)$ ) of a (a) Lambertian point source (b) Lambertian object. . . . .	16
Figure 3.3	Spectral ROS of a Lambertian point source; (a) in $\omega_x\omega_u$ subspace (b) in $\omega_y\omega_v$ subspace. . . . .	19
Figure 3.4	Spectral ROS of a Lambertian object; (a) in $\omega_x\omega_u$ subspace (b) in $\omega_y\omega_v$ subspace. . . . .	19
Figure 3.5	Theoretical 4-D spectrum of an LF plotted as $k_{x_0,y_0}$ slices . . . . .	21
Figure 3.6	Spectrum of Wheat & Silos LF from EPFL LF dataset. . . . .	21
Figure 3.7	Spectral energy of LFs (a) 98% of total energy (b) 95% of total energy . . . . .	22
Figure 3.8	Selective filtering . . . . .	23
Figure 3.9	Algorithm flow diagram . . . . .	24
Figure 3.10	2-D filters (a) parallelogram passband (b) circular passband . . . . .	28

Figure 3.11 SAIs of LFs from EPFL dataset for denoising with $\sigma = 0.2$ . From left, first column: original image, second column: noisy image, third column: output of proposed method, fourth column: output of 4-D hyperfan filter, fifth column: output of 4-D planar filter, (a)-(e) Diplodocus (f)-(j) Graffiti (k)-(o) Reeds. . . . .	34
Figure 3.12 SAIs of LFs from EPFL dataset for denoising with $\sigma = 0.2$ . From left, first column: original image, second column: noisy image, third column: output of proposed method, fourth column: output of 4-D hyperfan filter, fifth column: output of 4-D planar filter, (a)-(e) Diplodocus (f)-(j) Graffiti (k)-(o) Reeds. . . . .	35
Figure 4.1 The spectral ROS of a Lambertian object and the passband of the 4-D hyperfan filter $H(\mathbf{z})$ (a) in the $\omega_x\omega_u$ subspace; (b) in the $\omega_y\omega_v$ subspace. . . . .	37
Figure 4.2 The structure of the proposed 4-D sparse FIR hyperfan filter.	38
Figure 4.3 The magnitude response of $H_{xu}(\mathbf{z})$ (a) with nonsparse coefficients; (b) with sparse coefficients. . . . .	41
Figure 4.4 (a) NRMSE between the frequency responses of the sparse $H_{xu}(\mathbf{z})$ and the nonsparse $H_{xu}(\mathbf{z})$ ; (b) number of nonzero coefficients of the sparse $H_{xu}(\mathbf{z})$ compared to the nonsparse $H_{xu}(\mathbf{z})$ ; w.r.t. $\theta$ and $h_{th}$ with $\alpha = 50^\circ$ . . . . .	42
Figure 4.5 (a) NRMSE between the frequency responses of the sparse $H_{xu}(\mathbf{z})$ and the nonsparse $H_{xu}(\mathbf{z})$ ; (b) number of nonzero coefficients of the sparse $H_{xu}(\mathbf{z})$ compared to the nonsparse $H_{xu}(\mathbf{z})$ ; w.r.t. $\alpha$ and $h_{th}$ with $\theta = 15^\circ$ . . . . .	42
Figure 4.6 (a) NRMSE between the frequency responses of the sparse $H_{xu}(\mathbf{z})$ and the nonsparse $H_{xu}(\mathbf{z})$ ; (b) number of nonzero coefficients of the sparse $H_{xu}(\mathbf{z})$ compared to the nonsparse $H_{xu}(\mathbf{z})$ ; w.r.t. $\alpha$ and $\theta$ with $h_{th} = 0.01$ . . . . .	42
Figure C.1 (a) Bowtie-shaped passband (b) Enlarged portion of (a) . . . . .	60
Figure D.1 SAIs of LFs from EPFL dataset for denoising with $\sigma = 0.1$ . From left, first column: original image, second column: noisy image, third column: output of proposed method, fourth column: output of 4-D hyperfan filter, fifth column: output of 4-D planar filter, (a)-(e) Diplodocus (f)-(j) Graffiti (k)-(o) Houses & Lake (p)-(t) Reeds (u)-(y) Rolex Learning Center. . . . .	63

Figure D.2	SAIs of LFs from EPFL dataset for denoising with $\sigma = 0.3$ . From left, first column: original image, second column: noisy image, third column: output of proposed method, fourth column: output of 4-D hyperfan filter, fifth column: output of 4-D planar filter, (a)-(e) Diplodocus (f)-(j) Graffiti (k)-(o) Houses & Lake (p)-(t) Reeds (u)-(y) Rolex Learning Center. . . . .	64
Figure D.3	SAIs of Houses & Lake LF for denoising with $\sigma = 0.2$ . (a) original image, (b) noisy image, (c) output of proposed method, (d) output of 4-D hyperfan filter, (e) output of 4-D planar filter . . . . .	65
Figure D.4	SAIs of Rolex Learning Center LF for denoising with $\sigma = 0.2$ . (a) original image, (b) noisy image, (c) output of proposed method, (d) output of 4-D hyperfan filter, (e) output of 4-D planar filter . . . . .	65

## List of Tables

3.1	Chosen parameter values . . . . .	27
3.2	Output comparison for grayscale LFs over a range of noise levels .	31
3.3	Average result comparison for grayscale LFs . . . . .	31
3.4	Output comparison for color LFs over a range of noise levels . . .	32
3.5	Average result comparison for color LFs . . . . .	32
4.1	The mean and standard deviation of NRMSE and number of non-zero coefficients for the instances considered in figures 4.4, 4.5 and 4.6. . . . .	43
4.2	The means and the standard deviations of the SSIM indices between the volumetric refocused images obtained with the proposed sparse and the nonsparse hyperfan filters. . . . .	43
4.3	Volumetric refocused images obtained with the proposed sparse filter and the non-sparse filter . . . . .	45
4.4	Volumetric refocused images obtained with the proposed sparse filter for $\alpha = 60^\circ$ and $\alpha = 105^\circ$ . . . . .	46
4.5	Output comparison for color noisy LFs with noise variance $\sigma^2=0.02$	47

## List of Abbreviations

<b>1-D</b>	One-Dimensional
<b>2-D</b>	Two-Dimensional
<b>3-D</b>	Three-Dimensional
<b>4-D</b>	Four-Dimensional
<b>5-D</b>	Five-Dimensional
<b>7-D</b>	Seven-Dimensional
<b>AWGN</b>	Additive White Gaussian Noise
<b>BRDF</b>	Bidirectional Reflectance Distribution Function
<b>CDS</b>	Correlated Double Sampling
<b>CNN</b>	Convolutional Neural Network
<b>DFT</b>	Discrete Fourier Transform
<b>DOF</b>	Depth of Field
<b>EPI</b>	Epipolar Plane Image
<b>FFT</b>	Fast Fourier Transform
<b>FIR</b>	Finite-Extent Impulse Response
<b>FPGA</b>	Field Programmable Gate Array
<b>HT</b>	Hard-Thresholding
<b>IBR</b>	Image-Based Rendering
<b>iid</b>	Independent and Identically Distributed
<b>LF</b>	Light Field
<b>NRMSE</b>	Normalized-Root-Mean-Square Error
<b>PSNR</b>	Peak-Signal-To-Noise Ratio
<b>ROS</b>	Region of Support
<b>SAI</b>	Sub-Aperture Image
<b>SSIM</b>	Structural Similarity



## List of Algorithms

3.1	Frequency domain 2-D filtering . . . . .	25
3.2	2-D overlap-add method . . . . .	26

## Chapter 1

### INTRODUCTION

Over the years, scientists have come up with models to describe and simulate the nature and propagation of light. One such model is the plenoptic function proposed by Adelson and Bergen [1], which fully describes the light rays emanating from a scene. The term “plenoptic” has been derived from the Latin term “plenus” meaning complete or full, and “optic” pertaining to vision. This seven-dimensional (7-D) function describes the intensity of light rays passing through a point at every possible location  $(x, y, z)$  at every possible angle  $(\theta, \phi)$ , for every possible wavelength  $\lambda$  at every time  $t$ , as shown in Figure 1.1. The function takes the form  $P(x, y, z, \theta, \phi, \lambda, t)$ .

A four-dimensional (4-D) light field (LF) is a simplified version of the plenoptic function derived by using the following assumptions and conditions [2, 3],

- Intensity of a light ray remains the same along its direction of propagation within free space (with no occlusion);
- Consider red, green and blue (RGB) wavelengths only;
- Consider an instant in time, i.e., static scenes.

These conditions eliminate the third spatial dimension  $z$ , wavelength dimension  $\lambda$  and time dimension  $t$ , to give us a function of the form  $l(x, y, u, v)$ , which is called an LF.

A 4-D LF represents light rays emanating from a scene, as a function of both position and direction [2, 4]. An LF is richer in information compared to a two-dimensional (2-D) image, since an LF consists of 2-D positional as well as 2-D angular information, whereas a 2-D image consists only of 2-D angular information. In addition, the internal structure of LF cameras enable enhanced light gathering for an aperture size similar to that of a conventional camera [5]. Consequently, LFs facilitate larger depth of field (DOF), post-capture refocusing and depth estimation [6]. Hence, LF cameras have gained prominence in recent years

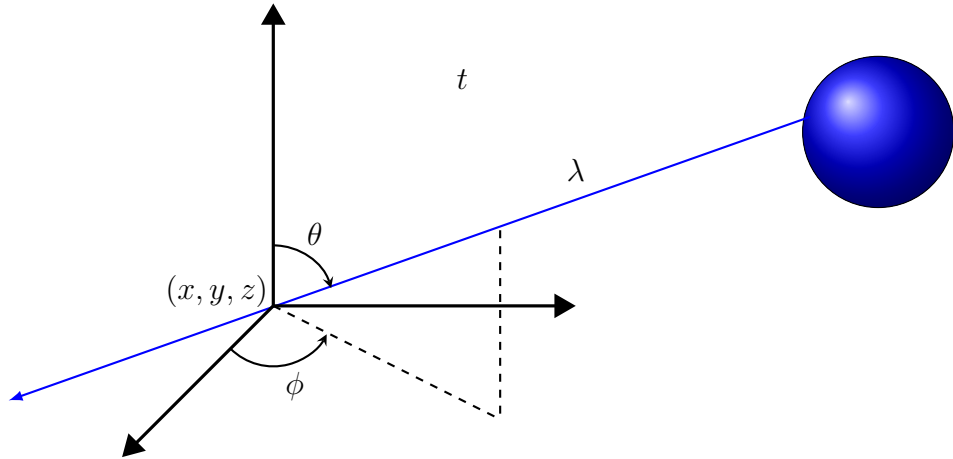


Figure 1.1: 7-D plenoptic function

as a powerful tool for imaging in challenging environments such as underwater, at night and in fog or smoke, where increasing illumination is not always practical [7, 8]. Here, the extended DOF in LF cameras help overcome the tradeoff between aperture size and DOF in conventional cameras, by collecting more light for the same DOF.

Similar to 2-D imaging, LFs are also susceptible to noise, which in turn affects the subsequent processing. Hence, a required step in the LF processing pipeline is to denoise them. In the literature, there are multiple LF denoising algorithms that are being used to resolve this issue. However, partly due to the immensity of LFs and partly because of the complexity of the algorithms, the denoising process is too slow and resource consuming, to use LF processing in real-time.

One of the key applications of LFs is post-capture refocusing, where the focus of an image is altered after the image is captured by the camera. This phenomenon, which was not possible with 2-D imaging, has been made feasible with LFs. This idea was first demonstrated in [6], where planar refocusing has been achieved with shifting and averaging in the spatial domain. Recently, an algorithm for refocusing with wider depth of field than that can be obtained from planar refocus, is proposed in [9], where the method is referred to as volumetric refocusing.

In this thesis, the main objective is to propose novel algorithms that have reduced complexity, thereby achieving near real-time light field processing. The main contributions of the thesis are presented in the next subsection.

## 1.1 Contributions of the Thesis

In this thesis, two computationally efficient algorithms are proposed for LF processing. The first of the proposed algorithms is a low-complexity LF denoising algorithm. Here, we utilize the sparsity of the region of support (ROS) of an LF in the 4-D frequency domain and propose selective filtering in the 4-D frequency domain. Furthermore, utilizing the separability of multidimensional Fourier transform, we employ a mixed-domain approach, where the LF is processed in the spatial domain for spatial (x,y) dimensions and in the frequency domain for the angular (u,v) dimensions. The LF in the mixed domain is next filtered using 2-D overlap-add method [10, ch. 2.2] and 2-D discrete Fourier transform (DFT) with fast Fourier transform (FFT) algorithms, in order to reduce the computational and memory complexity. Experimental results prove that the proposed method runs in 3 seconds, whereas the available denoising methods take minutes to execute.

The second contribution of this thesis is a sparse filter for volumetric refocusing of LFs. We extend the work in [9] and propose a cascade of two 4-D hyperfan filters, by utilizing the partial separability of the ROS of an LF. Since the spectral regions encompassed by the two filters are independent of each other, the designs of the two filters reduce to designs of 2-D fan filters. Furthermore, we utilize the hard-thresholding approach to obtain sparse filter coefficients for the two fan filters. The volumetric refocused image is obtained for the central sub-aperture image (SAI) of the LF by spatial domain filtering using partial-difference equations. The theoretical and experimental results demonstrate that the proposed sparse filter achieves 72% reduction of computational complexity with the degradation of accuracy by only 1%. Part of this work had been published in [11].

## 1.2 Outline of the Thesis

The remaining chapters of this thesis are organized as follows:

- Chapter 2: Review on Light Field Imaging

In this chapter, LFs are reviewed in detail including LF acquisition and applications of LFs. Section 2.5 discusses the current methods utilized for LF denoising and their limitations in terms of complexity. LF refocusing and volumetric refocusing of LFs are reviewed in Section 2.6.

- **Chapter 3: A Low-Complexity Denoising Algorithm for Light Fields**  
This chapter presents the proposed low-complexity LF denoising algorithm. In Section 3.3 the ROS of the spectrum of an LF is elaborated. Section 3.4 analyzes the sparsity of LFs in the frequency domain and provides the background for the proposed method. In Section 3.5, the proposed low-complexity algorithm is explained in detail. Experimental results and complexity analysis are presented in Section 3.7, where denoising performance and complexity of the proposed method are compared with those of the previously reported linear methods [3, 12].
  
- **Chapter 4: A 4-D Sparse Filter for Volumetric Refocusing of Light Fields**  
This chapter presents the proposed 4-D sparse hyperfan filter for volumetric refocusing of LFs. Section 4.2 briefly analyzes volumetric refocusing of LFs. In section 4.3, the proposed 4-D sparse filter is discussed in detail. Section 4.4 presents the outputs of the proposed filter and analyzes the distortion caused by sparsity of the filter. Furthermore, it compares the computational complexity of the proposed sparse filter with that of the non-sparse hyperfan filter [9].
  
- **Chapter 5: Conclusions and Future Work**  
This chapter presents the concluding remarks and directions for future work.

## Chapter 2

### REVIEW ON LIGHT FIELD IMAGING

We review the LF acquisition and processing with related concepts, and applications of LFs in this chapter. First we review the 7-D plenoptic function and its simplified versions derived by utilizing different assumptions. Secondly we elaborate about LFs, their applications and types of LF cameras. Next we evaluate the types of noise that can be found in LFs and the available denoising methods that are currently being used to attenuate noise. Furthermore, we evaluate the computational and memory complexity of the available methods. In the next section, LF refocusing and volumetric refocusing are discussed in detail where complexity of available method is also evaluated.

#### 2.1 Plenoptic Function

The 7-D plenoptic function proposed by Adelson and Bergen [1] describes all the radiant energy perceived from the point of view of the observer rather than the point of view of the source [13]. As stated in Chapter 1, the plenoptic function describes the intensity of all light rays passing through a point at every possible location  $(x, y, z)$ , at every possible angle  $(\theta, \phi)$ , for every possible wavelength  $\lambda$  at every time  $t$ . This results in the following form of the plenoptic function.

$$p = P(x, y, z, \theta, \phi, \lambda, t) \quad (2.1)$$

In order to reduce the data size of the plenoptic function, we restrain its viewing space by utilizing assumptions, a thorough study about which is presented in [14]. As mentioned previously in Chapter 1, the following assumptions are used to derive the 4-D LF from the plenoptic function.

- Radiance along a light ray through empty space remains a constant until it meets an occlusion. Hence, we limit our interest to the light rays leaving the convex hull of a bounded scene. This means we no longer need to measure

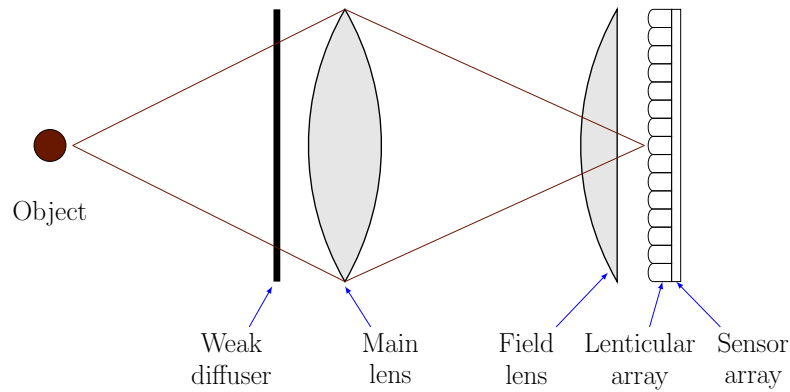


Figure 2.1: Camera design by Adelson in [17]

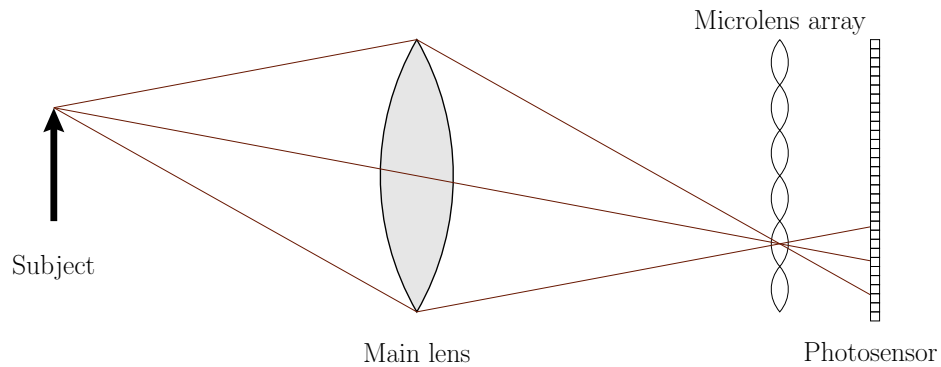


Figure 2.2: Camera design by Ng et al. in [18]

the radiance at different positions along each light ray. Hence, measuring radiance can be constrained to an arbitrary surface surrounding the scene.

- Wavelength dimension is simplified into three channels namely, Red-Green-Blue (RGB). Each channel gives the integration of  $p$  (2.1) over a certain range of frequencies (i.e., wavelength). All mathematical operations are conducted separately on each channel.
- Scene is static. Hence, we can drop time dimension from 2.1.

Except the 4-D LF, there are other simplified versions of the plenoptic function that are derived by utilizing a lesser or higher number of assumptions as required. For example, in the case of five-dimensional LF videos [14–16], only the first two from above assumptions are used (i.e., scene is dynamic).

## 2.2 Light Fields and Applications

As mentioned in Chapter 1, LFs are richer in information content compared to 2-D conventional images. Additionally, LF cameras also enable enhanced light

gathering for the same DOF, compared to that in a conventional camera. Consequently, LFs have emerged as powerful tools for achieving results otherwise impossible or limited.

The first application where LFs were used is image-based rendering (IBR) [2]. The traditional approach to rendering has been to model the scene geometrically and use images to map textural information onto the model [19]. The traditional model-based rendering requires a huge processing power to make the rendered scene photo-realistic [13]. IBR overcomes the challenge of photorealism by using a large collection of images of the same scene from multiple viewpoints. Since SAIs of LFs provide multiple views of the same scene, LFs are useful in IBR.

Post-capture refocusing which was proposed in [6], is one of the applications that has been made possible with LFs. As the name suggests, this allows generating multiple views of the same scene, each having different focus, after capturing the LF [20]. Therefore, unlike in 2-D conventional imaging where, images focused on different depths have to be captured separately, an LF has to be captured once only. Post-capture refocusing is also termed as *synthetic aperture photography*.

Another application where LFs are used is occlusion removal. The redundant information captured by LFs enable enhancing objects based on their depth. Using depth filters on LFs, occlusions that obstruct the required view can be attenuated [3, 21–23]. This is extremely useful in monitoring and surveillance systems.

Since LF cameras provide enhanced light gathering compared to a conventional camera, LFs have overcome the traditional trade-off between aperture size and DOF. When aperture size is increased, it enables collecting more light but DOF reduces and vice versa. In contrast with the conventional cameras, LF camera collects more light for the same aperture size and DOF. Hence, LF cameras have been extremely useful in photographing dimly lit areas such as underwater or foggy environments [5, 7].

Using LFs for virtual reality has been explored in [24]. Moreover, some works have explored utilizing LF technology in the field of robotics [25–28]. [25] proposes three closed-form solutions to estimate the six degree of freedom trajectory of a camera. While this method uses an array of cameras with fixed parameters, the work in [26] focuses on reducing the generated amount of data to allow real-time processing, without degrading the visual odometer’s performance. In addition to these works in plenoptic camera based visual odometry, [27] proposes a complete framework that includes a calibration process as well. Furthermore, [28] investigates utilizing LFs in on-orbit service robots, in order to have extended





Figure 2.3: Lytro Illum light field camera (Source-<https://www.lytro.com/illum/>)



Figure 2.4: Raytrix light field camera (Source-<http://lightfield-forum.com/raytrix/raytrix-r11-3d-lightfield-camera/>)

DOF, and generate three-dimensional (3-D) depth maps.

Additionally, recent works have explored using LFs and plenoptic cameras in automotive safety [29], ear recognition [30], volumetric particle image velocimetry [31], and presentation attack detection in facial recognition systems [32].

### 2.3 Light Field Acquisition

Over the years a variety of devices have been presented for the purpose of acquisition of LFs. In 1992 Adelson and Wang [17] have proposed a design for a plenoptic camera, which is shown in Figure 2.1. Although it simplified the assembly and calibration of the LF camera, it was still too long to be portable [33].

In 2005, Ng et al. implemented a rather simplified design which allowed the design prototype to be portable (See Figure 2.2). This made hand-held LF photography possible [6,18]. Ng went ahead and founded Lytro [34] in 2006, which in 2012 launched the first consumer grade LF camera. Recently Lytro released a more sophisticated LF camera named Illum [35], which is shown in 2.3. However, the first commercially available LF camera was launched by Raytrix [36] in 2009. As opposed to Lytro, Raytrix targets industrial and scientific applications rather than consumer grade applications. One of their LF cameras is shown in 2.4.

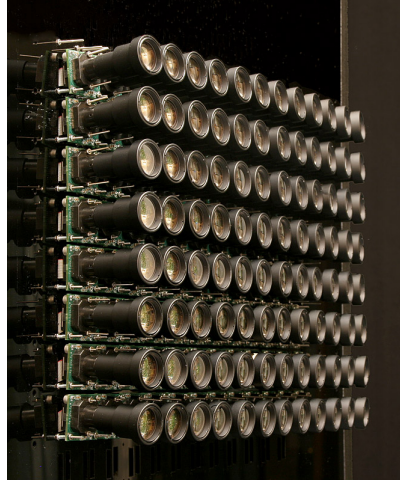


Figure 2.5: Stanford multi-camera array (Source-<http://graphics.stanford.edu/projects/array/>)

Apart from these devices, Computer Graphics Laboratory in Stanford University have their own setups for the acquisition of LFs. The multi-camera array (Figure 2.5) [37] which was built by Bennett Wilburn can be considered as the first among them. The development of this camera is described in detail in [38]. Other than that, they have also come up with an LF microscope [39] and a spherical gantry [40]. The first is dedicated to studying LFs at a microscopic scale while the latter is utilized as a device for acquiring fully surrounded LFs of small objects. A camera design small enough to be integrated into mobile devices has been proposed in [41]. This module, named *PiCam*, is a  $4 \times 4$  camera array which creates a  $4 \times 4$  array of images, each with  $1000 \times 750$  pixels.

## 2.4 Noise in Light Fields

Similar to 2-D conventional imaging, noise impacts LF imaging as well, which in turn affects the subsequent processing. Types of noise added by image sensors into these media can be specified as follows [42, 43]:

- $kT/C$  noise (Reset noise) –  $kT/C$  noise is the thermal noise which is generated due to resetting of pixels after their readout. This type of noise can be suppressed in image sensors by using a correlated double sampling (CDS) circuit [44]. Furthermore, it is modeled as additive white Gaussian noise (AWGN);
- Thermal noise (Johnson noise/Nyquist noise/Johnson–Nyquist noise) - Thermal noise is generated due to the random nature of electrons. It is modeled as AWGN since it has zero mean and a flat, wide bandwidth [43];

- Flicker noise (1/f noise) – Flicker noise is caused by carrier fluctuations. At low frequencies this can be the dominant noise component, whereas at higher frequencies, thermal noise become dominant over 1/f noise. Furthermore, 1/f noise is mostly suppressed by CDS circuit in image sensors [44];
- Shot noise – Shot noise occurs due to the variation in the amount of photons collected by the image sensor, during exposure time. It follows a Poisson distribution in general, and except at very low intensity levels, its distribution approximates a Gaussian distribution. This type of noise may become prominent in photographs captured at low intensity;
- Quantization noise – This type of noise occurs due to the discretization of pixel values.

## 2.5 Denoising Light Fields

From a broad perspective, the available LF denoising techniques can be categorized into three groups; image or video denoising algorithms extended for LFs, linear methods and non-linear methods specifically developed for LFs. Since an LF can be represented as a 2-D array of 2-D images, any off-the-shelf image denoising algorithm can be used to denoise each 2-D image named SAI, individually. Image denoising algorithms which can be utilized for this naive approach are summarized in [45] and [46]. The common theme in best-performing image denoising algorithms, is to utilize block matching [47], or formulate the denoising task as an optimization problem, with a regularization penalty such as total variation [48] or sparsity [49].

However, since these methods consider 2-D slices of the 4-D LF, they do not fully utilize the redundancies existing in LFs. To further employ the interdependencies of LFs, it is possible to use traditional video denoising algorithms such as V-BM3D [50] and V-BM4D [51], by re-arranging SAIs into an image sequence and filtering along this additional dimension. Another approach is to use epipolar plane image (EPI) representation of LFs, where an EPI is a 2-D slice of 4-D LF, obtained by keeping one angular and one spatial dimension constant. Same as with SAIs, EPIs also can be stacked and filtered as a video, utilizing the available video denoising methods. In [52], Li et al. propose a two-staged framework, where each EPI is separately filtered using total-variation image denoising method proposed in [48]. Noisy LF is filtered by forming horizontal EPIs in the

first stage of the algorithm, whereas in the second stage, intermediate result from the first stage is filtered as vertical EPIs. However, this method considers only the spatial redundancy in EPIs. In [53], Sepas-Moghaddam et al. propose a method that utilizes temporal redundancy as well. There, SAIs are mapped into an ordered sequence by snake-like row-major scan, which is next converted to an EPI sequence in the following step and filtered using V-BM4D algorithm [51].

The second category consists of the linear denoising methods specifically developed for LFs, which are the 4-D planar [3] and the 4-D hyperfan filters [12]. Characteristics of LFs in the spatial frequency domain have been explored in [12], where it has been derived that the spectral ROS of an LF, is a 4-D hyperfan. Hence, a 4-D hyperfan-shaped filter in the spatial frequency domain is used to segregate the LF signal from noise.

The latest LF denoising techniques proposed in [54] and [55], belong to the category of nonlinear methods specifically developed for LFs. LFBM5D filter [54] extends the BM3D method [47] by considering the redundancies occurring in 2-D angular dimensions of the LF. In [55], Chen et al. propose a framework with two sequential convolutional neural network (CNN) modules. Based on anisotropic parallax features, the first CNN denoted as *syn-Net* is used to construct the parallax structure of the LF, whereas a second CNN denoted as *view-Net* is used to restore non-parallax details onto the output of *syn-Net*.

### 2.5.1 Complexity of Light Field Denoising Methods

Although there is a plethora of methods for LF denoising, they are too computationally expensive to be implemented in real-time scenarios, which require fast execution times. Filtering each SAI independently, requires time equal to the number of SAIs in the LF times the execution time for denoising one SAI. This is a huge drawback especially regarding the optimization-based image denoising methods such as BM3D and learning-based techniques such as k-SVD [49]. Similarly, filtering each EPI separately requires time proportional to the dimensions of the LF.

Compared to filtering each SAI or EPI separately, filtering a SAI or EPI sequence using video denoising methods such as V-BM3D or V-BM4D is a better approach since the execution is faster. However, it is also not appropriate for a real-time application, since block-matching methods generally have longer execution times in the order of minutes. In contrast to this, filtering in the frequency domain using the 4-D hyperfan or 4-D planar filter is faster since they can be implemented using fast algorithms such as the FFT. However, hyperfan filter

and the planar filter require manual parameter tuning for the best performance, which is unsuitable for real-time processing. In addition, these methods tend to demand a huge amount of additional memory for processing, since a 4-D filter and 4-D DFT of the LF, have to be loaded into memory for computation. The five-dimensional (5-D) approach taken to employ redundancy of LFs [54], exacerbate the curse of dimensionality, thereby making it more computationally and memory intensive. As reported in [56], the LFBM5D method [54] takes about 7 hours on an octo-core processor. Furthermore, recent LF denoising methods that employ CNNs, also focus on best denoising performance rather than real-time processing. With the recent advancement into utilizing LFs in field robotics [57], there is a requirement for fast and simpler algorithms that are able to run on battery-powered devices, without putting too much computational load. However, a little work has been done so far on real-time LF denoising. Hence, there is a need for simpler, less resource-hungry methods for LF denoising.

## 2.6 Light Field Refocusing

As stated above, an LF consists of 2-D positional as well as 2-D angular information as opposed to conventional 2-D imaging, which has only angular information. This additional information can be employed to accomplish post-capture refocusing, which is also identified as synthetic aperture photography. This allows the user to generate multiple views with different focal planes, after the LF is captured. In contrast, changing the focal plane of a conventional 2-D photograph after capturing, has only recently been made possible using neural networks [58, 59].

This possibility in LFs was first demonstrated in [6], where images having different focal planes have been obtained from the same LF, by shifting the LF with respect to the angular dimensions and averaging with respect to the positional dimensions. A frequency domain implementation of LF refocusing was proposed in [60], where a refocusing method based on generalized Fourier slice theorem has been implemented using FFT algorithms. Moreover, [61] achieves refocusing through depth-adaptive splatting. All these methods achieve planar-refocusing, in that the generated views have a single depth in focus.

However, putting a depth range in focus is desirable when the scene is non-planar. Recently, a filter which achieves a wider DOF (referred to as *volumetric refocusing*), was proposed in [9]. Here, volumetric refocusing is attained by using a 4-D filter having a hyperfan-shaped passband in the 4-D frequency domain.

### 2.6.1 Complexity of Volumetric Refocusing of Light Fields

The key difference in the hyperfan filter method for volumetric refocusing is that its output is a 4-D LF whereas in the planar refocusing methods, the output is a 2-D image. This leads to higher computational and memory complexity in volumetric refocusing, since the number of output samples of the hyperfan filter is huge, compared to the size of a 2-D image. As an instance, a typical LF of the EPFL LF dataset [62] is  $15 \times 15 \times 625 \times 434$ . Output of the hyperfan filter will be of the same dimensions, whereas the outputs of planar refocusing methods will be of size  $625 \times 434$ .

## Chapter 3

# A LOW-COMPLEXITY DENOISING ALGORITHM FOR LIGHT FIELDS

### 3.1 Introduction

In this chapter, we propose a novel low-complexity LF denoising algorithm, by utilizing the sparsity of LFs in the frequency domain. To this end, we first analyze the spectrum of a light field that corresponds to a Lambertian object. A Lambertian object is assumed to be comprised of Lambertian surfaces. A Lambertian surface is one that scatters incoming light uniformly in all directions. In other words, it is a surface which diffusely reflects light. In terms of bidirectional reflectance distribution function (BRDF), Lambertian surfaces are those that have a constant BRDF [63, ch. 2.2]. Such a Lambertian surface can be modeled as a collection of Lambertian point sources. Therefore, the LF of a Lambertian object is obtained by the superposition of the LFs of its point sources. Furthermore, almost all natural materials except water, show diffuse reflectance [64]. Although natural scenes can have some specular components such as reflection on water, they primarily consist of diffusely reflected light. Hence, in our work, we adopt the assumption of a Lambertian scene. Consequently, in the analysis presented in this chapter, we focus on the LF representation of a Lambertian point source, and its spectra.

We begin the analysis by reviewing the ROS of the spectrum of a Lambertian point source, and showing that it is a plane through the origin in the 4-D frequency domain. Furthermore, it is shown that for a Lambertian point source, the ROS of its spectrum is given by the intersection of two 3-D hyperplanes. Next, we review the ROS of the spectrum of a Lambertian object, and show that it is a 3-D hyperfan in 4-D frequency domain [8, 12]. Based on this analysis, we infer that LFs have a sparse representation in the 4-D frequency domain.

Next we propose a technique named *selective filtering* where we eliminate redundant filtering processes by utilizing the *sparsity* of the LF signal in the 4-D frequency domain. As elaborated in Sections 3.5 and 3.7, we propose a novel

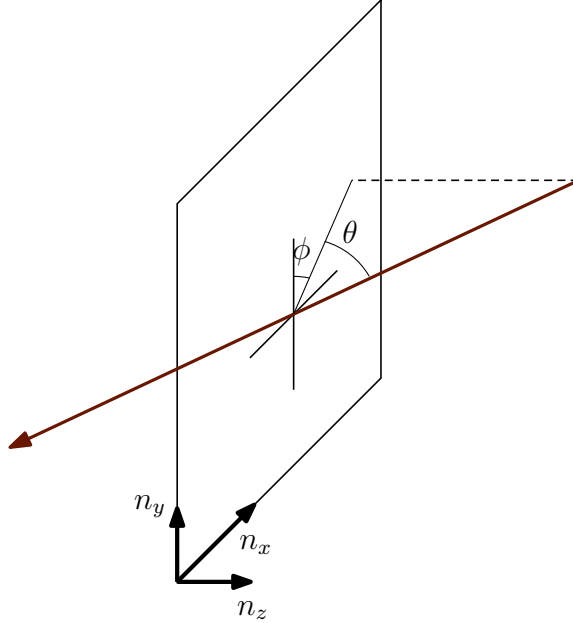


Figure 3.1: Spherical-cartesian parameterization.

low-complexity LF denoising algorithm which achieves LF denoising in near real-time. Subsequently, we present a complexity analysis of the proposed algorithm and the previously proposed linear methods [3, 12]. Finally, we draw conclusions by comparing the results of the proposed algorithm with those of the methods proposed in [3, 12] in terms of PNSR, SSIM and computational complexity.

## 3.2 Light Field Parameterization

### 3.2.1 Spherical-Cartesian Parameterization

Recall that, an LF is a 4-D representation of the plenoptic function. Pertaining to the assumptions stated in section 2.1, we will drop wavelength  $\lambda$ , depth  $z$  and time  $t$ , to denote the remaining function as  $L(x, y, \theta, \phi)$  [65]. The value of this function is a color triplet, since RGB channels are considered separately. This representation which is identified as *spherical-cartesian parameterization*, is depicted in Figure 3.1.

In spherical-cartesian parameterization, direction of light ray is represented by two angles  $\theta$  and  $\phi$ . These two angles depict the point of intersection of each ray, with a reference sphere centered at the coordinates  $(x, y)$ . Here,  $(x, y)$  coordinates lie on a reference plane which denotes the location of a particular light ray. The advantage of this notation is that light rays traveling in all directions are represented with equal resolution. However, because it uses angles to denote the direction, even the simplest calculations require the use of trigono-



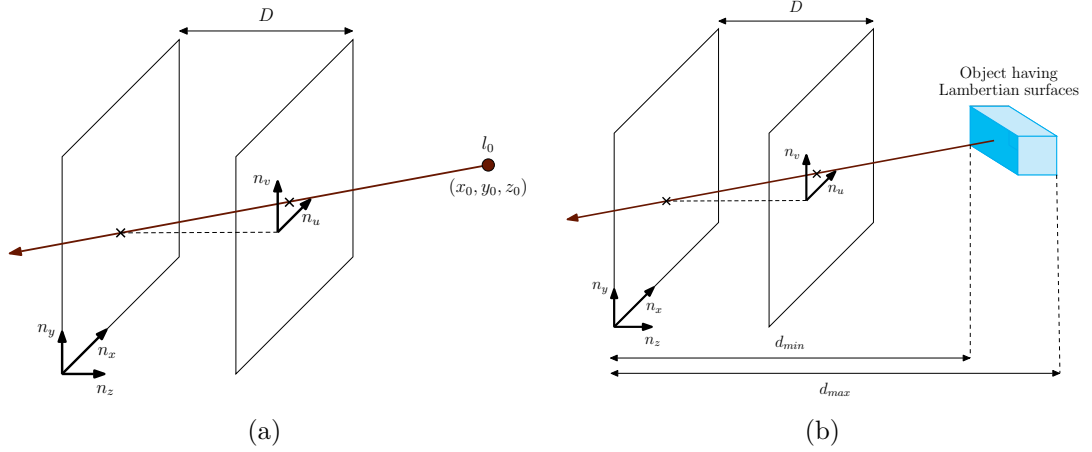


Figure 3.2: The two-plane parameterization (with the locally defined image coordinates  $(n_u, n_v)$ ) of a (a) Lambertian point source (b) Lambertian object.

metric functions, which make it quite complex. To overcome this issue, *two-plane parameterization* was introduced.

### 3.2.2 Two-Plane Parameterization

Instead of the reference sphere, we may use a second reference plane. Two sets of local coordinates are used to avoid confusion as  $(x, y)$  and  $(u, v)$  for each plane as shown in Figure 3.2(a). Ideally, the two reference planes are of infinite extent. However, in practical applications these two are sampled with finite ROS. Resulting function is denoted as  $L(x, y, u, v)$ . Similar to the case in spherical–cartesian parameterization, this function also gives values as color triplets.

With this representation, most operations turn out to be simpler than in spherical–cartesian parameterization. However, the drawback of this representation is that it only includes light rays propagating from the  $(u, v)$  plane towards  $(x, y)$  plane. This is a smaller subset of the plenoptic function than in the spherical–cartesian parameterization. Also, resolution of the representation varies with the direction of propagation. On the other hand, spherical–cartesian parameterization represents light rays traveling in all directions, with the same resolution.

Regardless of its weaknesses, considering its simplicity and the equivalence to the structure of a camera (lens and sensor), two-plane parameterization is widely employed in the analysis of light fields. Hence, throughout this thesis two–plane parameterization will be used.

### 3.3 Review of Light Field Spectrum

In this section we review the 4-D frequency spectrum of an LF. Consider the two-plane parameterization of a Lambertian point source and a Lambertian object depicted in Figure 3.2. Here the coordinates  $(n_x, n_y) \in \mathbb{Z}^2$  denote the camera plane whereas the coordinates  $(n_u, n_v) \in \mathbb{Z}^2$  denote the image plane. The constant distance between the two planes is denoted by  $D$ .

#### 3.3.1 LF Representation of a Lambertian Point Source

First we consider a discrete domain LF  $l_p(\mathbf{n})$ ,  $\mathbf{n} = (n_x, n_y, n_u, n_v) \in \mathbb{Z}^4$  corresponding to a Lambertian point source of intensity  $l_0$  located at depth  $z_0 \in \mathbb{R}^+$ , shown in Figure 3.2(a). By employing the EPI analysis [66], the Lambertian point source can be represented as a plane of constant value  $l_0$  given by the intersection of two 3-D hyperplanes [3],

$$mx + u + c_x = 0 \quad (3.1a)$$

$$my + v + c_y = 0, \quad (3.1b)$$

where

$$m = \frac{D}{z_0} \quad (3.2a)$$

$$c_x = \frac{-Dx_0}{z_0} \quad (3.2b)$$

$$c_y = \frac{-Dy_0}{z_0}. \quad (3.2c)$$

Hence, the light field  $l_p(\mathbf{n})$  comprising a Lambertian point source can be formulated as follows.

$$l_p(\mathbf{n}) = l_0 \delta(mx + u + c_x) \delta(my + v + c_y) \quad (3.3)$$

Using multi-dimensional Fourier transform theory, the spectrum of a Lambertian point source  $\mathcal{L}_p(\boldsymbol{\omega})$ ,  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_u, \omega_v) \in \mathbb{R}^4$ , can be obtained from (3.3) as [67],

$$\mathcal{L}_p(\boldsymbol{\omega}) = 4\pi^2 l_0 \delta(\omega_x - m\omega_u) \delta(\omega_y - m\omega_v) e^{j(c_x\omega_u + c_y\omega_v)} \quad (3.4)$$

### 3.3.2 ROS of the Spectrum

From (3.4), the ideal ROS  $\mathcal{R}_p$  of the spectrum  $\mathcal{L}_p(\boldsymbol{\omega})$  inside the Nyquist hypercube  $\mathcal{N}$ , is derived as [3, 68],

$$\mathcal{R}_p = \mathcal{H}_{xu} \cap \mathcal{H}_{yv}, \quad (3.5)$$

where,

$$\mathcal{H}_{xu} = \left\{ \boldsymbol{\omega} \in \mathcal{N} \mid \omega_x - \left( \frac{m\Delta x}{\Delta u} \right) \omega_u = 0 \right\} \quad (3.6a)$$

$$\mathcal{H}_{yv} = \left\{ \boldsymbol{\omega} \in \mathcal{N} \mid \omega_y - \left( \frac{m\Delta y}{\Delta v} \right) \omega_v = 0 \right\}. \quad (3.6b)$$

Here,  $m = D/z_0$ ,  $\Delta i$ ,  $i = x, y, u, v$ , is the sampling interval along the corresponding dimension.

Hence, spectral ROS  $\mathcal{R}_p$  is a plane through the origin in the 4-D frequency domain, that lies at the intersection of two 3-D hyperplanes

$$\omega_x - \left( \frac{m\Delta x}{\Delta u} \right) \omega_u = 0 \quad (3.7a)$$

$$\omega_y - \left( \frac{m\Delta y}{\Delta v} \right) \omega_v = 0 \quad (3.7b)$$

Since  $\frac{1}{m} = \tan \alpha$  (See Figure 3.3), we can obtain the following relationship for  $\alpha$  from (3.2a).

$$\alpha = \tan^{-1} \left( \frac{z_0}{D} \right) \quad (3.8)$$

From (3.8) we can conclude that, since  $D$  is a constant, the orientation  $\alpha$  of the two hyperplanes 3.7a and 3.7b, depends solely on the depth  $z_0$  of the point source.

By modeling the Lambertian object as a collection of Lambertian point sources having depth  $z_0 \in [d_{min}, d_{max}]$ , we can derive the spectral ROS  $\mathcal{R}_o$  of an LF corresponding to a Lambertian object (see Figure 3.2(b)). Thus,  $\mathcal{R}_o$  is given as the superposition of the spectral ROS of point sources, in the depth range  $[d_{min}, d_{max}]$ . Utilizing the linearity of the multidimensional Fourier transform [10, ch. 1.3],  $\mathcal{R}_o$  can be obtained as [22, 68]

$$\begin{aligned} \mathcal{R}_o &= \bigcup_{z_0} \mathcal{R}_p \\ &= \bigcup_{z_0} (\mathcal{H}_{xu} \cap \mathcal{H}_{yv}). \end{aligned} \quad (3.9)$$

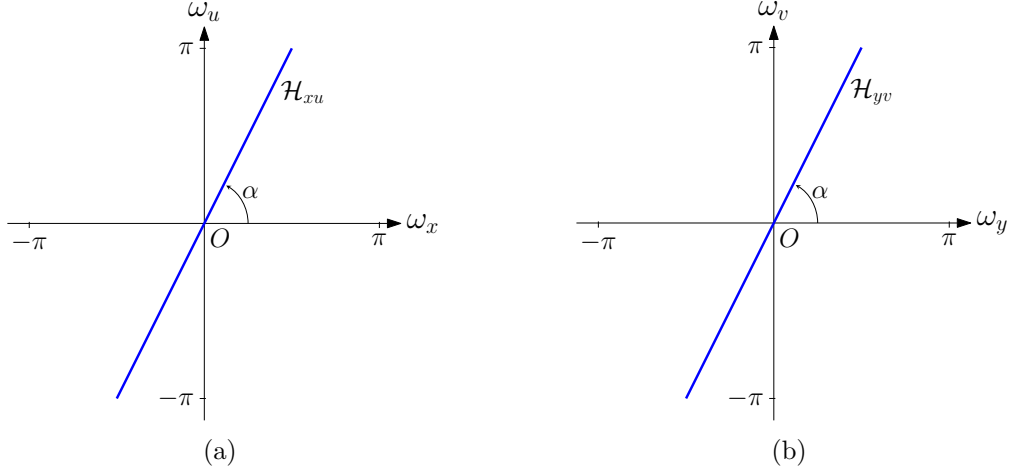


Figure 3.3: Spectral ROS of a Lambertian point source; (a) in  $\omega_x\omega_u$  subspace (b) in  $\omega_y\omega_v$  subspace.

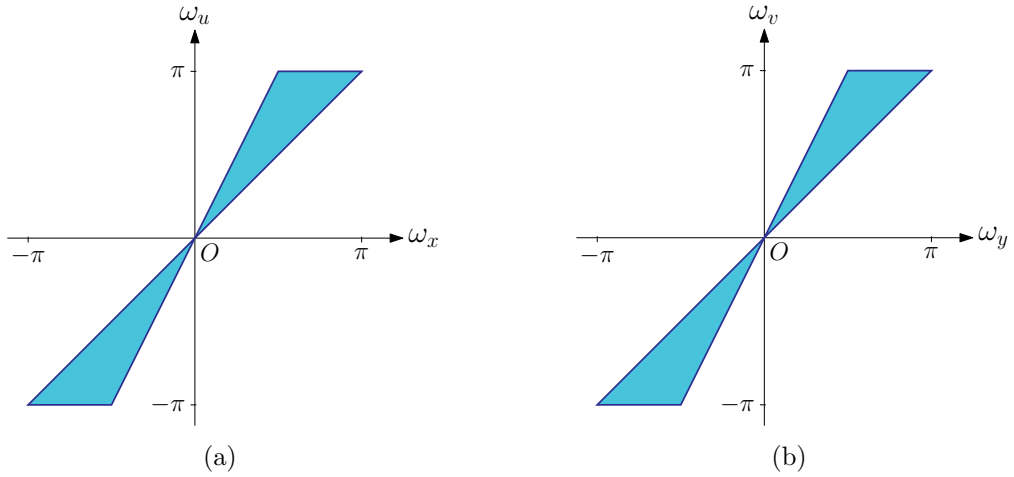


Figure 3.4: Spectral ROS of a Lambertian object; (a) in  $\omega_x\omega_u$  subspace (b) in  $\omega_y\omega_v$  subspace.

Figure 3.4 illustrates 2-D projections of the spectral ROS  $\mathcal{R}_o$  which is a hyperfan, where the angular width depends on the range of depths occupied by the Lambertian object [12].

### 3.3.3 Dimensionality Gap in Light Fields

As elaborated above, the scene depth corresponds to slope  $m$  in the LF through (3.2a). However, it has been observed that most slopes in the 4-D LF do not correspond to a physical depth, which is characterized by the following two constraints [9].

$$m_{min} < \frac{\omega_x}{\omega_u} < m_{max} \quad (3.10a)$$

$$m_{min} < \frac{\omega_y}{\omega_v} < m_{max} \quad (3.10b)$$

Here  $m_{max}$  and  $m_{min}$  denote the highest and lowest slopes of the fan shapes shown in Figure 3.4. The shape of the frequency spectrum defined by (3.10a) and (3.10b) is named the dual-fan [22]. Furthermore, the third constraint given below describes the Lambertian surfaces lying on planes in the 4-D frequency domain.

$$\frac{\omega_x}{\omega_u} = \frac{\omega_y}{\omega_v} \quad (3.11)$$

The shape of the frequency spectrum defined by (3.11) is identified as the hypercone [9]. Furthermore, [9] also shows that the frequency domain ROS of an LF, which is named hyperfan, is given by the intersection of the dual-fan and the hypercone. The ROS of an LF is referred to as the *dimensionality gap* in [69].

The dimensionality gap can be clearly observed in the 4-D frequency spectrum of an LF. We consider 2-D slices  $L_{\omega_{x_0}, \omega_{y_0}}(\omega_u, \omega_v)$  of the LF spectrum  $L$ , where the spatial frequencies  $\omega_x$  and  $\omega_y$  are held constant. We refer to these 2-D slices as  $k_{x_0, y_0}$  slices, following the convention in [69]. From (3.7a) and (3.7b) we can obtain the following relationship for the angular frequencies  $\omega_u$  and  $\omega_v$ , when the spatial frequencies  $\omega_x$  and  $\omega_y$  are held constant by letting  $\omega_x = \omega_{x_0}$  and  $\omega_y = \omega_{y_0}$ .

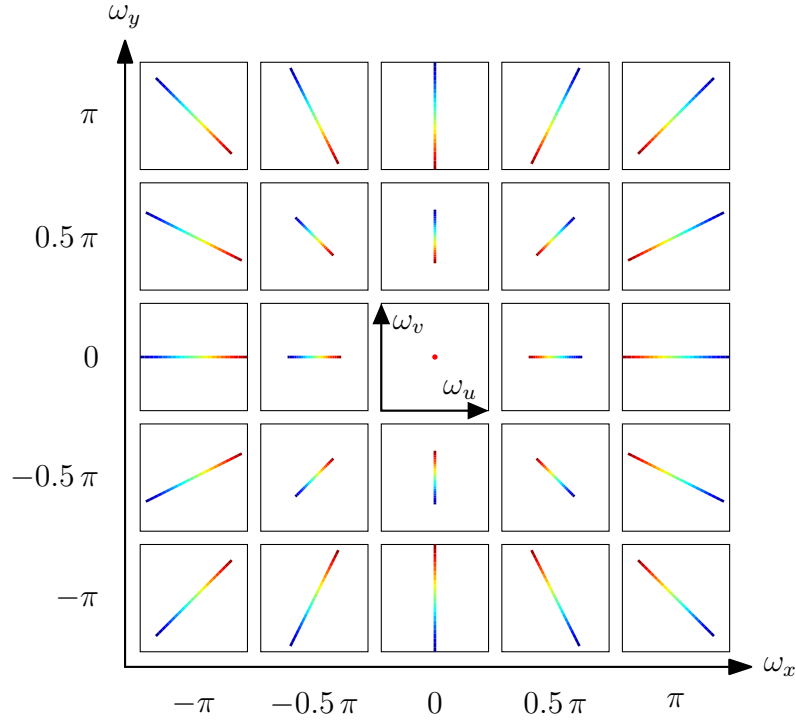
$$\omega_v = \left( \frac{\Delta_v \Delta_x \omega_{y_0}}{\Delta_u \Delta_y \omega_{x_0}} \right) \omega_u \quad (3.12)$$

Since  $\omega_{x_0}$ ,  $\omega_{y_0}$  and  $\Delta_i$ ,  $i = x, y, u, v$  are constants, the 4-D spectrum follows a linear relationship in the 2-D  $k_{x_0, y_0}$  slices. The theoretical 4-D spectrum obtained as a 2-D array of 2-D  $k_{x_0, y_0}$  slices, using (3.12), is shown in Figure 3.5. Here, we have considered an LF with  $5 \times 5$  array of SAIs.

### 3.4 Sparsity and Selective Filtering

By observing the LF spectrum plotted in Figure 3.5, it can be seen that the spectral energy of the LF of a Lambertian object, is limited to a small portion in the 4-D Nyquist hypercube. Hence, the light field has a sparse representation in the frequency domain. However, the spectrum of an actual LF, taken from an LF camera, somewhat differs from the ideal spectrum due to the finite field of view of cameras and the finite number of SAIs. This can be illustrated by drawing samples from the EPFL LF dataset [62, 70] and plotting their  $k_{x_0, y_0}$  slices. Their spectra were observed to take the shape shown in Figure 3.6. From the spectra, it is evident that LFs are sparse low-pass signals in the 4-D frequency domain.

This can be further demonstrated by plotting only a fraction of spectral energy of the LF, in discrete 4-D frequency domain. Figure 3.7 shows the spectra of LFs

Figure 3.5: Theoretical 4-D spectrum of an LF plotted as  $k_{x_0, y_0}$  slices

from EPFL LF dataset when 98% and 95% of total spectral energy is plotted respectively. These two plots have been obtained by plotting spectra for a sample of 40 LFs drawn from the EPFL LF dataset, and superimposing the plots for each percentage considered. The plots in Figure 3.7 clearly demonstrate the sparsity of LFs in the frequency domain. Therefore, noise can be substantially attenuated by filtering only the  $k_{x_0, y_0}$  slices closer to the center of the spectrum, where most of the signal energy is present, and setting the spectrum to zero in the remaining  $k_{x_0, y_0}$  slices. We identify this process as *selective filtering* in the discrete 4-D frequency domain.

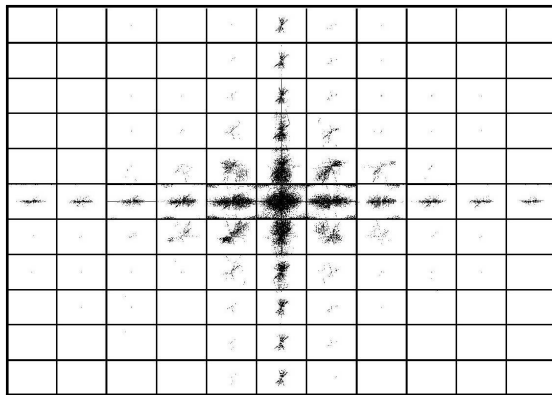


Figure 3.6: Spectrum of Wheat &amp; Silos LF from EPFL LF dataset.

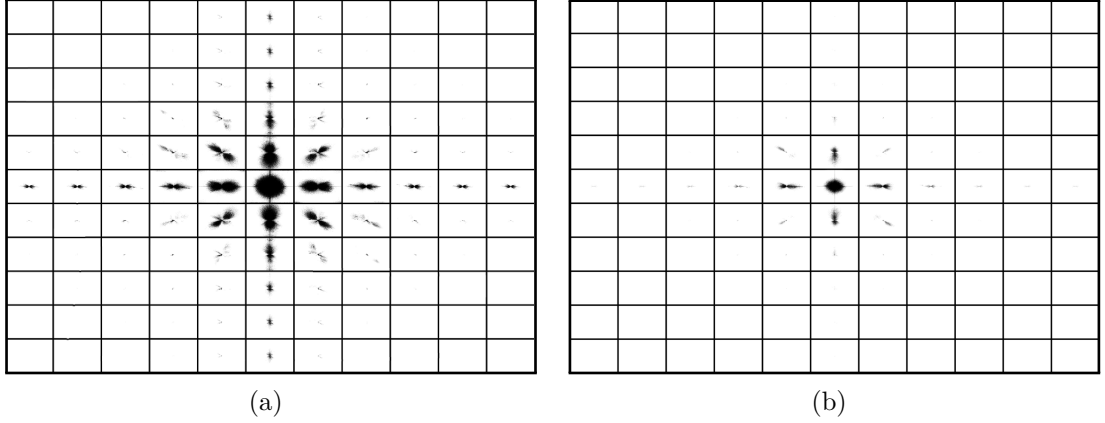


Figure 3.7: Spectral energy of LFs (a) 98% of total energy (b) 95% of total energy

### 3.5 Proposed Low-complexity Algorithm

The proposed low-complexity algorithm is derived by utilizing the sparsity of the LFs in discrete 4-D frequency domain, which was elaborated in the previous section. By observing the spectra of multiple LFs from the EPFL dataset and the Stanford dataset [71], we could distinguish the following characteristics.

- The  $k_{x_0, y_0}$  slice in the center of the spectrum closely follows a circular shape. This corresponds to the subset of  $\boldsymbol{\omega}$  where  $\omega_x = 0$  and  $\omega_y = 0$ . However, in the center  $k_{x_0, y_0}$  slice of the theoretical 4-D spectrum shown in Figure 3.5, spectrum is available for only a small subset of  $\omega_u$  and  $\omega_v$ .
- Other  $k_{x_0, y_0}$  slices take the shape of a parallelogram, and the spectrum declines in amplitude as  $\omega_x$  and  $\omega_y$  increases (i.e. as we move away from the center). In contrast, the theoretical 4-D spectrum shown in Figure 3.5, does not show a similar declining relationship.

Employing these characteristics of the spectrum, we propose to use parallelogram filters and a circular filter in the 2-D frequency domain  $\omega_u \omega_v$  to attenuate noise.

The 4-D Fourier transform is a *separable operator*, since it can be performed as a sequence of one-dimensional (1-D) Fourier transforms, one on each of the four dimensions [10]. Therefore, as an alternative to converting the 4-D LF to the discrete 4-D frequency domain, we consider the LF in what we specify as the *mixed domain*, where conversion to frequency domain is done only in  $n_x$  and  $n_y$  dimensions. The output of this procedure is a 4-D signal of which  $n_x$  and  $n_y$  dimensions are in the frequency domain, whereas  $n_u$  and  $n_v$  dimensions are in the spatial domain. Furthermore, utilizing the sparsity of the LF in the 4-D frequency domain, selective filtering is employed as shown in Figure 3.8, by filtering only

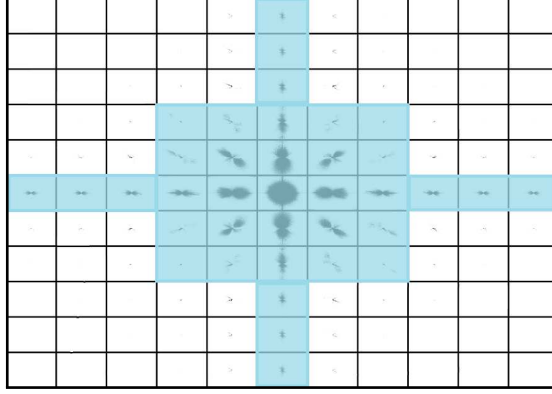


Figure 3.8: Selective filtering

the  $k_{x_0, y_0}$  slices highlighted in blue. The remaining slices are replaced with zeros since the spectral energy present in those slices is infinitesimal compared to that in the highlighted slices.

These techniques are utilized in order to reduce the time complexity of the proposed algorithm by eliminating the requirement to apply 4-D DFT to LF. In addition to this, 2-D *overlap-add* method is used for filtering each  $k_{x_0, y_0}$  slice as an attempt to reduce the additional memory required for denoising.

Figure 3.9 depicts the steps of the proposed algorithm, which can be enumerated as follows:

1. Convert the noisy LF to mixed domain signal  $LF_{in}(k_x, k_y, n_u, n_v)$  by applying the 2-D DFT to  $n_x$  and  $n_y$  dimensions. Here, the LF has  $N_x \times N_y$  SAIs. Furthermore,  $k_x = 0, 1, \dots, N_x - 1$  and  $k_y = 0, 1, \dots, N_y - 1$ .
2. For each selected  $k_{x_0, y_0}$  slice (highlighted in Figure 3.8):
  - (a) Use 2-D *overlap-add* method to get filtered output for each  $k_{x_0, y_0}$  slice. For this purpose we use the derived filter impulse response  $h$  corresponding to the selected  $k_{x_0, y_0}$  slice. For  $(k_x, k_y) = (0, 0)$  circular passband filter is used. For all other instances parallelogram filter is used. First, the filter impulse response is converted to its frequency response by applying 2-D DFT in  $u$  and  $v$  dimensions.
  - (b) Next, we consider a small square block from  $k_{x_0, y_0}$  slice. This block is filtered by point-wise multiplication with the filter frequency response.
  - (c) Finally, the output block is converted back to mixed domain by 2-D IDFT in  $n_u$  and  $n_v$  dimensions, and saved.
  - (d) Similarly, whole  $k_{x_0, y_0}$  slice is filtered in a block-wise manner.



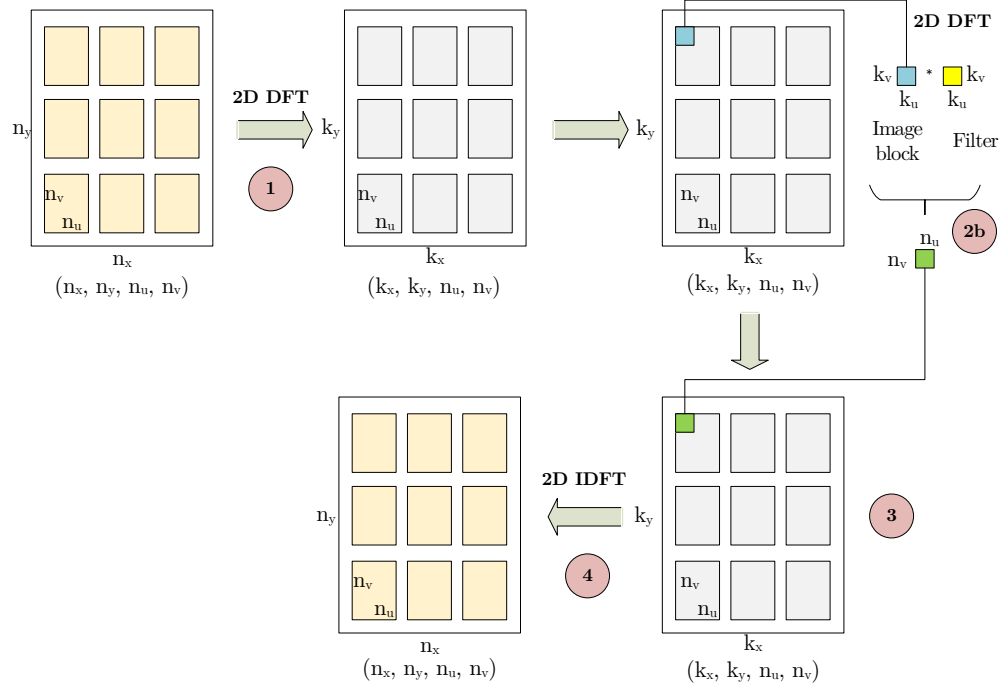


Figure 3.9: Algorithm flow diagram

3. Remaining  $k_{x_0, y_0}$  slices are set to zero. The filtered output  $LF_{out}(k_x, k_y, n_u, n_v)$  is obtained by processing each  $k_{x_0, y_0}$  slice as mentioned in the step above.
4. Apply 2-D inverse DFT to  $LF_{out}(k_x, k_y, n_u, n_v)$  on  $n_y$  and  $n_x$  dimensions to get denoised LF,  $lfdn(n_x, n_y, n_u, n_v)$ .

Pseudocode representation of the proposed algorithm is presented in algorithms 3.1 and 3.2. Selection of parameters including bandwidth  $B$  in line 20 of algorithm 3.1 and block size  $L$  in algorithm 3.2, is explained in Section 3.6.1.

### 3.6 2-D Filter Design

The procedure of designing the 2-D parallelogram and circular filters is elaborated in this section. As mentioned above, the circular filter is used for filtering the central  $k_{x_0, y_0}$  slice  $(k_x, k_y) = (0, 0)$ , whereas parallelogram filters are used for the remaining slices. By observing the spectra in Figures 3.6 and 3.7, it can be noticed that spectrum grows thinner as it moves away from the central  $k_{x_0, y_0}$  slice. Following this realization, parallelogram filter bandwidth  $B$  is chosen as follows.

$$B = l \times \frac{\pi}{\sqrt{r_x^2 + r_y^2}} \quad (3.13)$$

**Algorithm 3.1** Frequency domain 2-D filtering**Require:** Noisy LF  $lf$ **Ensure:** Denoised LF  $lfdn$ 


---

```

1: procedure LFFILTER( $lf$ )
2:    $[N_x, N_y, N_u, N_v] \leftarrow size(lf)$ 
3:    $M_u \leftarrow$  length of filter  $u$  dimension
4:    $M_v \leftarrow$  length of filter  $v$  dimension
5:    $F_u \leftarrow M_u + N_u - 1$  ▷ fft length
6:    $F_v \leftarrow M_v + N_v - 1$  ▷ fft length
7:    $LF_{in} \leftarrow$  2D fft of LF in  $y$  and  $x$  dimensions
8:   for  $i \leftarrow 1 : N_y$  do
9:     for  $j \leftarrow 1 : N_x$  do
10:      if  $j \leq (N_x + 1)/2$  then
11:         $x = j - 1$ 
12:      else
13:         $x = j - N_x - 1$ 
14:      end if ▷ x coordinate
15:      if  $i \leq (N_y + 1)/2$  then
16:         $y = 1 - i$ 
17:      else
18:         $y = N_y + 1 - i$ 
19:      end if ▷ y coordinate
20:       $B \leftarrow l \times \frac{\pi}{\sqrt{|x|^2 + |y|^2}}$  ▷ Bandwidth
21:      if  $x = 0$  and  $y = 0$  then
22:         $\alpha \leftarrow circular\ filter$ 
23:      else if  $x = 0$  then
24:         $\alpha \leftarrow 90$ 
25:      else if  $y = 0$  then
26:         $\alpha \leftarrow 0$ 
27:      else
28:         $\alpha \leftarrow \tan^{-1}(y/x)$ 
29:      end if
30:       $h_{uv} \leftarrow$  impule response of 2-D planar filter in  $u, v$ 
31:       $IM_{in} \leftarrow LF_{in}(i, j, :, :)$  ▷ (i,j)th slice of  $LF_{in}$ 
32:       $LF_{out}(i, j, :, :) \leftarrow OVERLAP\_ADD(IM_{in}, h_{uv})$ 
33:    end for
34:  end for
35:   $lfout \leftarrow$  4D inverse fft of  $LF_{out}$ 
36:   $lfdn \leftarrow$  remove transient of  $lfout$ 
37:  return  $lfdn$ 
38: end procedure

```

---

---

**Algorithm 3.2** 2-D overlap-add method

---

```

1: function OVERLAP_ADD( $im, h$ )
2:    $[im_{d1}, im_{d2}] \leftarrow size(im)$ 
3:    $[K, M] \leftarrow size(h)$ 
4:    $L \leftarrow 220$ 
5:    $r1 \leftarrow$  remainder of  $im_{d1}$  after division by  $L$ 
6:    $r2 \leftarrow$  remainder of  $im_{d2}$  after division by  $L$ 
7:    $im1 \leftarrow im$  zero padded by  $L - r1$  and  $L - r2$ 
8:    $im_{d1} \leftarrow$  (length of  $im1$ )/ $L$ 
9:    $im_{d2} \leftarrow$  (width of  $im1$ )/ $L$ 
10:   $h \leftarrow h$  zero padded by  $L - 1$  on both sides
11:   $H \leftarrow$  2-D fft of  $h$  of size  $L + M - 1$  on both sides
12:  for  $i \leftarrow 1 : im_{d1}$  do
13:     $rw \leftarrow ((j - 1) * L + 1 : j * L)$ 
14:    for  $j \leftarrow 1 : im_{d2}$  do
15:       $cl = ((i - 1) * L + 1 : i * L)$ 
16:       $temp \leftarrow im1(rw, cl)$ 
17:       $X \leftarrow temp$  zero padded by  $M - 1$  on both sides
18:       $X \leftarrow$  zero centred 2-D fft of  $X$  of size  $L + M - 1$  on both sides
19:       $Y \leftarrow X \circ H$  ▷ Point-wise multiplication
20:       $y \leftarrow$  2-D inverse fft of  $Y$ 
21:      Save filtered block to relevant position of output image  $out$ 
22:    end for
23:  end for
24:  remove zero pad
25:  return  $out$ 
26: end function

```

---

Here,  $l$  is the parameter that allows varying the parallelogram passband bandwidth, whereas  $\sqrt{r_x^2 + r_y^2}$  is the geometric distance of a particular  $k_{x_0, y_0}$  slice from the slice in the center.

Derivation of the ideal infinite-extent impulse response of the parallelogram filter and the circular filter are given in Appendix A and Appendix B respectively. To obtain the finite-extent impulse response, we utilize the windowing method with a Hamming window as follows.

$$h_{uv}(n_u, n_v) = h_{uv}^I(n_u, n_v)w_{uv}(n_u, n_v) \quad (3.14)$$

where,  $h_{uv}(n_u, n_v)$  is the finite-extent impulse response,  $h_{uv}^I(n_u, n_v)$  is the infinite-extent impulse response, and  $w_{uv}(n_u, n_v)$  is the window function.

### 3.6.1 Parameter Selection

From the derived infinite-extent filter responses in Appendix A and Appendix B, it is evident that there are multiple parameters that need to be tuned for the best denoising performance achievable by the proposed method. Parallelogram filter has two distinct parameters, namely, cutoff frequency  $C_v$  (See Figure 3.10(a)) and filter bandwidth  $l$  (3.13). Circular passband filter has a single distinct parameter, which is the cutoff frequency  $\omega_c$  (See Figure 3.10(b)). Filter order  $M$  and block size  $L$  for 2-D overlap-add method (Algorithm 3.2), are common to both passband shapes.

Since there are multiple filter parameters, this becomes a multi-variable optimization problem with an objective function to maximize peak-signal-to-noise ratio (PSNR) and structural similarity (SSIM). The optimized parameter values obtained through exhaustive search, are specified in Table 3.1.

Table 3.1: Chosen parameter values

Parameter	Value
$C_v$	0.7
$l$	0.2
$\omega_c$	0.7
$M$	61
$L$	220

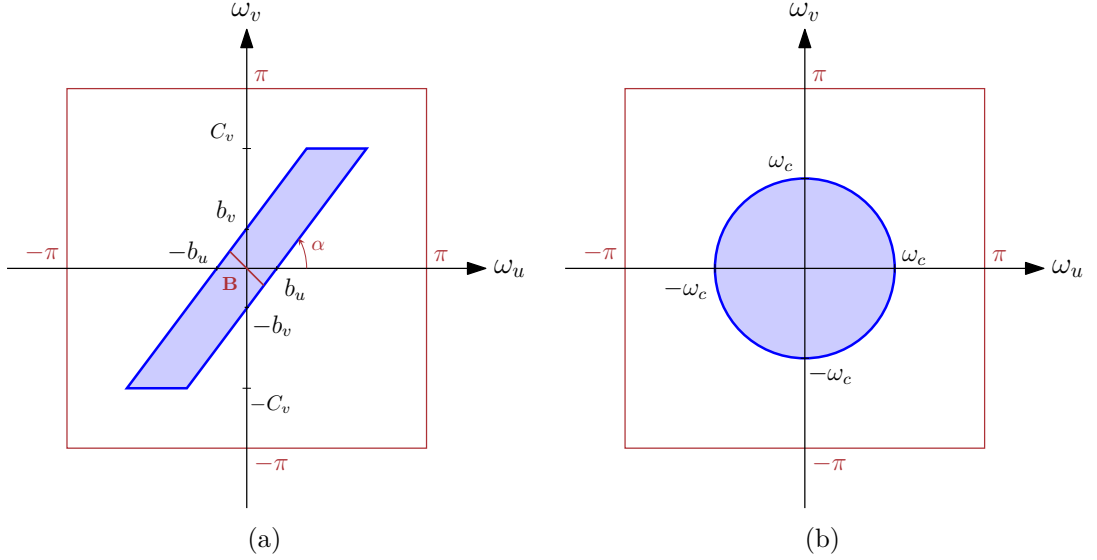


Figure 3.10: 2-D filters (a) parallelogram passband (b) circular passband

### 3.7 Results and Comparison

In the first part of this section, we theoretically compare the complexity of the proposed algorithm with the available methods, using the closed form expressions for complexity. In addition to this, we also consider an example case and provide a comparison of the number of computations and additional memory requirement. However, learning-based and block matching methods are excluded in this comparison, since we cannot exactly assess the number of computations or the additional memory requirement.

Experimental results of the proposed algorithm is compared with that of the previously reported linear filtering algorithms in this section. Denoising performance is compared in terms of both PSNR and SSIM. Moreover, execution time for each algorithm is recorded as an indication of its computational complexity.

#### 3.7.1 Complexity Analysis

The objective of this work has been to reduce the computational complexity and memory complexity of LF denoising, which was detailed in Section 2.5.1. Therefore, in this section, complexity of the proposed filter is compared with those of alternative methods.

Assuming an LF of dimensions  $N_x \times N_y \times N_u \times N_v$ , the total number of computations of the proposed filter can be derived as,

$$2N_x N_y \log(N_x N_y) + N_{sel} (3M_u M_v \log(M_u M_v) + M_u M_v) \quad (3.15)$$

where  $N_{sel}$  is the number of selected  $k_{x_0,y_0}$  slices for the step 2 of the proposed algorithm (Figure 3.9) and  $M_u, M_v$  are the FFT lengths for  $n_u$  and  $n_v$  dimensions, with  $M_u = M_v = L + M - 1$ . Here, the first term corresponds to the FFT and inverse FFT operations on  $n_x$  and  $n_y$  dimensions of the LF. The second term denotes the number of operations when 2-D overlap-add method is employed for the selected  $k_{x_0,y_0}$  slices.

For 4-D hyperfan [12] and 4-D planar filters [3], the total number of arithmetic operations can be specified as,

$$2N_xN_yN_uN_v \log(N_xN_yN_uN_v) + 2N_xN_yN_uN_v \quad (3.16)$$

As an example case, considering an LF of size  $17 \times 17 \times 1024 \times 1024$ , multiplications and additions per output sample for the hyperfan filter, totaled up to 52 and 158 respectively. For the same LF dimensions and parameters selected in section 3.6.1, the proposed method takes only 9 multiplications and 39 additions per sample.

Considering the additional memory requirement, the memory complexity of the proposed filter with double-precision buffers is given as,

$$16N_xN_yN_uN_v + 32M_uM_v \quad (3.17)$$

whereas the 4-D hyperfan implementation in [12] has a memory complexity of,

$$24N_xN_yN_uN_v \quad (3.18)$$

However, in order to overcome the computational and memory complexity of the 4-D hyperfan in [12], Dansereau et al., [9] suggest using a hybrid of the spatial-domain and the frequency-domain implementations. The key thing to note here is that, in software implementations, frequency-domain computations are faster but memory intensive, whereas spatial-domain computations require less memory, at the cost of slower performance. To overcome this trade-off, Dansereau et al., [9] suggests incorporating overlap-add and overlap-save methods.

For the same example LF size of  $17 \times 17 \times 1024 \times 1024$ , frequency-domain implementation of 4-D hyperfan proposed in [12], consumes about 7273 MB for double-precision buffers. In [9], it is stated that the hybrid implementation takes about 603 MB for single-precision. Consequently, in double-precision arithmetic, the hybrid implementation of the 4-D hyperfan consumes 1206 MB. Comparatively, the software implementation of the proposed filter consumes about 4850

MB for double-precision buffers.

### 3.7.2 Experimental Results

To demonstrate the denoising performance, we test our algorithm on the EPFL LF dataset [62]. This dataset contains LFs of  $15 \times 15$  SAIs, with a resolution of  $625 \times 434$ . Before commencing the evaluation process, we eliminate the border images of these LFs which have been distorted by vignetting. The resized LFs contain only  $11 \times 11$  SAIs.

The experimental procedure can be described as follows:

#### 1. Add Gaussian noise

Since these noise types in 2.4 are predominantly Gaussian, we use Gaussian noise in our denoising experiments. The ground-truth is perturbed with independent and identically distributed (iid) Gaussian noise as given by,

$$l_n(\mathbf{n}) = l(\mathbf{n}) + \eta_{0,\sigma^2}(\mathbf{n}) \quad (3.19)$$

where  $l_n$  is the noisy LF,  $l$  is the ground-truth, and  $\eta_{0,\sigma^2}$  is iid zero mean white Gaussian noise, with standard deviation  $\sigma$ . In this analysis, different  $\sigma$  values are considered, ranging from 1% to 100%.

#### 2. Denoise LF

The noisy LF generated in the above step, is denoised using the method proposed in Section 3.5, and performance is compared with linear denoising methods proposed in [12] and [3].

#### 3. Performance criterion

Denoised LF is compared with the ground truth using PSNR and SSIM metric. In addition, algorithm execution time is recorded for proposed method and linear denoising methods proposed in [12] and [3], as a performance measure of time complexity.

The output PSNR and SSIM values obtained by following this procedure on grayscale LFs, are displayed in Tables 3.2 and 3.3. Tables 3.4 and 3.5 demonstrate the same performance metrics for color LFs. These tabulated results compare the denoising performance of the proposed algorithm with that of two alternative *linear* LF denoising methods, namely, hyperfan filter [12] and planar filter [3], for multiple noise levels ( $\sigma$ ). Here, the outputs from the hyperfan filter and the planar filter have been obtained using the optimized parameter values. Color SAIs of 3 LFs chosen from the EPFL dataset, are depicted in Figure 3.12 with

Table 3.2: Output comparison for grayscale LFs over a range of noise levels

LF	$\sigma$	Noisy LF		Proposed		Hyperfan		Planar	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Rolex	0.1	20.44	0.2670	<b>32.12</b>	<b>0.8606</b>	30.54	0.7388	28.70	0.8371
Learning Center	0.2	14.46	0.0702	<b>28.17</b>	<b>0.7265</b>	26.90	0.5756	26.61	0.7258
	0.3	10.85	0.0333	<b>23.83</b>	0.5930	22.91	0.4224	23.31	<b>0.6065</b>
Houses & Lake	0.1	20.50	0.1535	<b>35.19</b>	0.8644	31.43	0.7309	33.84	<b>0.8863</b>
	0.2	14.25	0.0454	<b>28.85</b>	0.7049	27.37	0.5508	28.69	<b>0.7509</b>
	0.3	10.67	0.0190	24.16	0.5591	23.18	0.3876	<b>24.26</b>	<b>0.6157</b>
Reeds	0.1	20.23	0.1575	<b>35.02</b>	0.8403	32.43	0.7195	34.97	<b>0.8626</b>
	0.2	14.03	0.0456	30.08	0.6903	28.38	0.5486	<b>30.58</b>	<b>0.7353</b>
	0.3	10.62	0.0197	25.55	0.5591	24.28	0.3945	<b>25.97</b>	<b>0.6151</b>
Graffiti	0.1	20.29	0.2827	<b>32.44</b>	0.8616	30.70	0.8029	31.49	<b>0.8669</b>
	0.2	14.09	0.1031	<b>28.72</b>	0.7585	27.65	0.6720	28.56	<b>0.7810</b>
	0.3	10.66	0.0487	24.78	0.6529	23.79	0.5307	<b>24.88</b>	<b>0.6853</b>
Diplodocus	0.1	20.07	0.1453	<b>34.36</b>	0.8367	29.78	0.6854	31.56	<b>0.8541</b>
	0.2	13.55	0.0450	<b>30.55</b>	0.6693	27.73	0.4978	29.35	<b>0.7147</b>
	0.3	10.32	0.0217	<b>27.18</b>	0.5373	24.77	0.3538	26.66	<b>0.5925</b>

Table 3.3: Average result comparison for grayscale LFs

	$\sigma = 0.1$			$\sigma = 0.2$			$\sigma = 0.3$		
	PSNR	SSIM	Time(s)	PSNR	SSIM	Time(s)	PSNR	SSIM	Time(s)
Noisy	20.30	0.2012	-	14.08	0.0619	-	10.62	0.0285	-
Proposed	<b>31.89</b>	0.8615	<b>3.36</b>	<b>28.38</b>	0.7334	<b>3.33</b>	<b>24.13</b>	0.6174	<b>3.33</b>
Hyperfan	29.80	0.7577	10.86	26.83	0.6140	11.07	23.13	0.4688	10.99
Planar	30.15	<b>0.8622</b>	10.23	27.33	<b>0.7578</b>	10.37	23.96	<b>0.6498</b>	10.36

results obtained for  $\sigma = 0.2$ , for visual comparison. Output SAIs for grayscale LFs are depicted in Figure 3.11. Supplementary results are provided in Appendix D.

Here, denoising performance is compared only with the linear methods since non-linear methods using optimization or learning-based techniques consistently outperform linear filters in the denoising task. For example, the best performing algorithms for 2-D images and 3-D videos are BM3D [47] and V-BM4D [51]. For LF denoising, so far the best denoising performance has been reported by LFBM5D method [54] and CNN based methods [72,73]. However, the objective of this work is to reduce the computational and memory complexity of LF denoising, thereby formulating a real-time algorithm for denoising LFs. Hence, performance of the proposed method is contrasted with that of other linear methods, which currently possess the least execution time for LF denoising.

From Table 3.2, it is evident that with the proposed filter, PSNR improvement ranges from 11 dB to 18 dB for grayscale images, whereas the SSIM improvement is between 0.55 to 0.75. The best output PSNR and SSIM values obtained by tuning the parameters of 4-D hyperfan and the 4-D planar filter are also indicated



Table 3.4: Output comparison for color LFs over a range of noise levels

LF	$\sigma$	Noisy LF		Proposed		Hyperfan		Planar	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Rolex	0.1	21.07	0.3680	<b>29.64</b>	<b>0.8556</b>	29.56	0.8418	29.04	0.8281
Learning Center	0.2	15.76	0.1676	<b>25.39</b>	<b>0.7244</b>	24.85	0.6661	24.53	0.6405
	0.3	12.72	0.0949	<b>21.60</b>	<b>0.6006</b>	21.11	0.5189	20.89	0.4900
Houses & Lake	0.1	21.24	0.5324	<b>31.25</b>	<b>0.8665</b>	30.59	0.8577	30.37	0.8498
	0.2	15.75	0.3644	<b>25.42</b>	<b>0.7417</b>	24.78	0.7075	24.58	0.6941
	0.3	12.67	0.2688	<b>21.42</b>	<b>0.6468</b>	20.94	0.6028	20.77	0.5886
Reeds	0.1	20.73	0.4151	<b>32.62</b>	0.8730	32.39	<b>0.8773</b>	32.00	0.8694
	0.2	15.35	0.1907	<b>27.15</b>	<b>0.7695</b>	26.39	0.7292	26.05	0.7094
	0.3	12.39	0.1078	<b>22.93</b>	<b>0.6681</b>	22.30	0.5981	22.04	0.5717
Graffiti	0.1	20.52	0.4564	30.24	0.8478	<b>30.99</b>	<b>0.8627</b>	30.62	0.8522
	0.2	15.11	0.2311	<b>26.71</b>	<b>0.7654</b>	26.25	0.7416	25.87	0.7193
	0.3	12.22	0.1379	<b>23.21</b>	<b>0.6856</b>	22.61	0.6337	22.30	0.6055
Diplodocus	0.1	20.29	0.6409	<b>31.19</b>	<b>0.9589</b>	30.58	0.9528	30.26	0.9491
	0.2	14.74	0.3620	<b>28.01</b>	<b>0.9129</b>	26.49	0.8840	26.08	0.8721
	0.3	11.90	0.2259	<b>24.69</b>	<b>0.8547</b>	23.34	0.8034	22.96	0.7845

Table 3.5: Average result comparison for color LFs

	$\sigma = 0.1$			$\sigma = 0.2$			$\sigma = 0.3$		
	PSNR	SSIM	Time(s)	PSNR	SSIM	Time(s)	PSNR	SSIM	Time(s)
Noisy	20.77	0.4826	-	15.34	0.2632	-	12.38	0.1671	-
Proposed	<b>30.99</b>	<b>0.8803</b>	<b>10.92</b>	<b>26.53</b>	<b>0.7828</b>	<b>10.93</b>	<b>22.77</b>	<b>0.6912</b>	<b>10.98</b>
Hyperfan	30.82	0.8785	16.32	25.75	0.7459	16.29	21.87	0.6314	16.37
Planar	30.46	0.8697	16.12	25.42	0.7270	16.09	21.79	0.6080	16.29

in the same table. Furthermore, average results for each noise level are juxtaposed in Table 3.3.

Similarly, from Table 3.4 we can observe that for color LFs the proposed method improves PSNR by a value between 9 dB to 14 dB. Moreover, we could also note that its SSIM improvement is between 0.3 and 0.6. In addition to this, Table 3.5 portrays the average results for each noise level, in the case of color LFs.

Comparing the output PSNR and SSIM values it is evident that the proposed method predominantly outperforms both the 4-D hyperfan and the 4-D planar filters.

The execution time for each method was measured using the MATLAB implementations on the same Windows 10 machine with an Intel i-7 processor. For both color and grayscale LFs, the proposed method achieved the best execution time. However, for color LFs, we can see an increase in execution time, nearly three times that for grayscale LFs. Considering non-linear LF denoising methods, filtering a grayscale LF as a SAI sequence with the V-BM4D algorithm takes more

than 5 minutes with the same experimental conditions. Similarly, the method proposed in [53] with V-BM4D chosen as the base denoising algorithm consumes around 8 minutes for grayscale LFs. Comparatively, the proposed method runs in near-real time.

However, there are a few limitations of the proposed LF denoising algorithm. Since the noise in the passband of the parallelogram and circular-shaped filters is not attenuated, there is still some residual noise on the output image. Although the effect of this residual noise is not visible in lower noise levels, for  $\sigma = 0.3$  and above, it can be slightly seen on the output images. Furthermore, since the proposed filter resembles a low-pass filter, blurring of edges is also present to some extent.

The proposed low-complexity LF denoising algorithm processes a typical LF in only 3 seconds and its denoising performance exceeds that of previously reported linear LF denoising methods. The proposed algorithm which is an approximation to the hyperfan passband, demonstrates this better performance since it omits filtering the parts of the spectrum with low signal energy, through selective filtering. The non-linear LF denoising methods consume minutes to complete, and yet they demonstrate superior denoising performance.

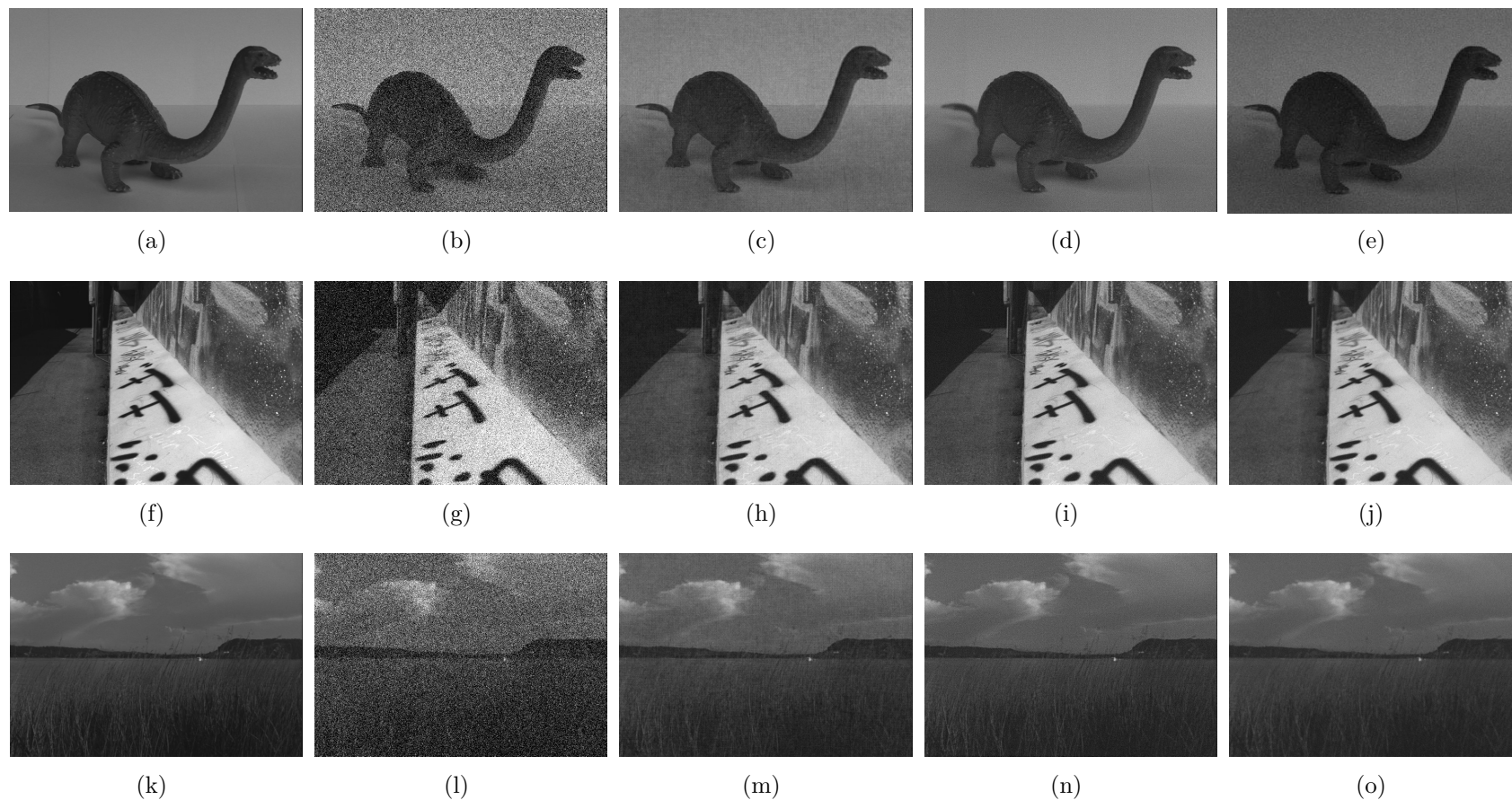


Figure 3.11: SAIs of LFs from EPFL dataset for denoising with  $\sigma = 0.2$ . From left, first column: original image, second column: noisy image, third column: output of proposed method, fourth column: output of 4-D hyperfan filter, fifth column: output of 4-D planar filter, (a)-(e) Diplodocus (f)-(j) Graffiti (k)-(o) Reeds.

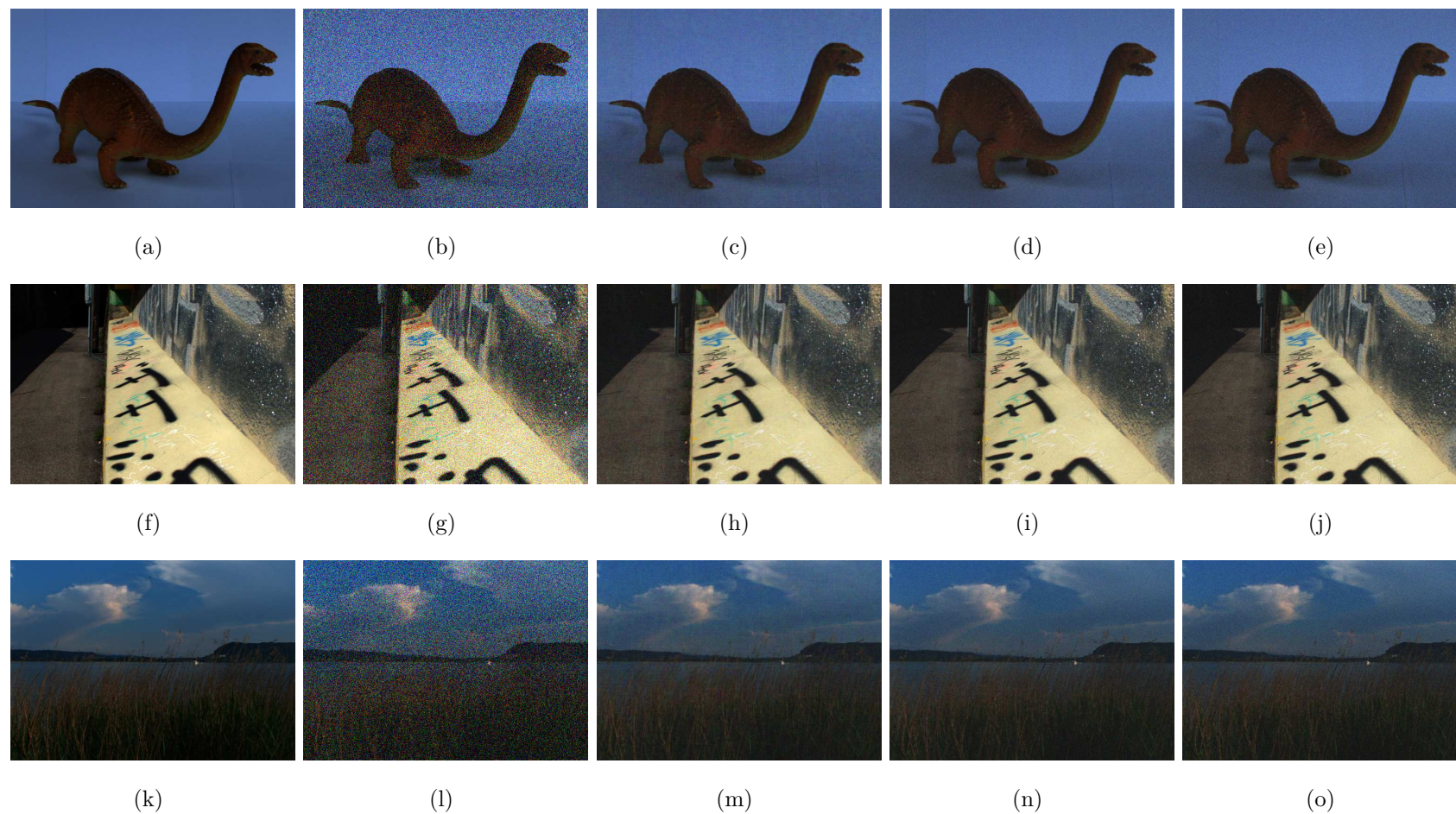


Figure 3.12: SAIs of LFs from EPFL dataset for denoising with  $\sigma = 0.2$ . From left, first column: original image, second column: noisy image, third column: output of proposed method, fourth column: output of 4-D hyperfan filter, fifth column: output of 4-D planar filter, (a)-(e) Diplodocus (f)-(j) Graffiti (k)-(o) Reeds.

## Chapter 4

# A 4-D SPARSE FILTER FOR VOLUMETRIC REFOCUSING OF LIGHT FIELDS

### 4.1 Introduction

In this chapter we analyze *post-capture refocusing* of LFs and propose a sparse 4-D finite-extent impulse response (FIR) hyperfan filter for volumetric refocusing of LFs. We first review the *volumetric focus*; a method that was proposed in [9] to maintain sharp focus over a range of depths selected by the user, rather than at a single depth as in *planar refocusing*. The 4-D hyperfan filter that is being employed for volumetric refocusing can also be used to denoise LFs, as we have previously discussed in Chapter 3.

Next, utilizing the *partial separability* of the spectral ROS and hard-thresholding (HT) approach, we propose a sparse filter that is designed as a cascade of two 4-D hyperfan filters. Note that, despite being simple, it has been shown that the HT approach is extremely effective in multidimensional sparse FIR filter designs employed for beamforming [74–76]. Subsequently, we compare the computational complexity of the proposed method with that of the non-sparse 4-D hyperfan filter. Finally, we compare the visual and analytical results of the two methods based on their MATLAB implementations.

### 4.2 Volumetric Refocusing

As elaborated in Chapter 2, post-capture refocusing is one of the novel tasks that has been made possible with 4-D LFs. This property was first demonstrated in [6] where images with different focal planes have been obtained by shifting and averaging the LF in the spatial domain. Planar refocusing based on generalized Fourier slice theorem has been proposed in [60]. Furthermore, a planar refocusing method using depth adaptive splatting has been achieved in [61]. However, it is desirable to put a volume in focus when dealing with non-planar scenes. *Volumetric focus* proposed in [9] achieves this task by employing a linear non-iterative

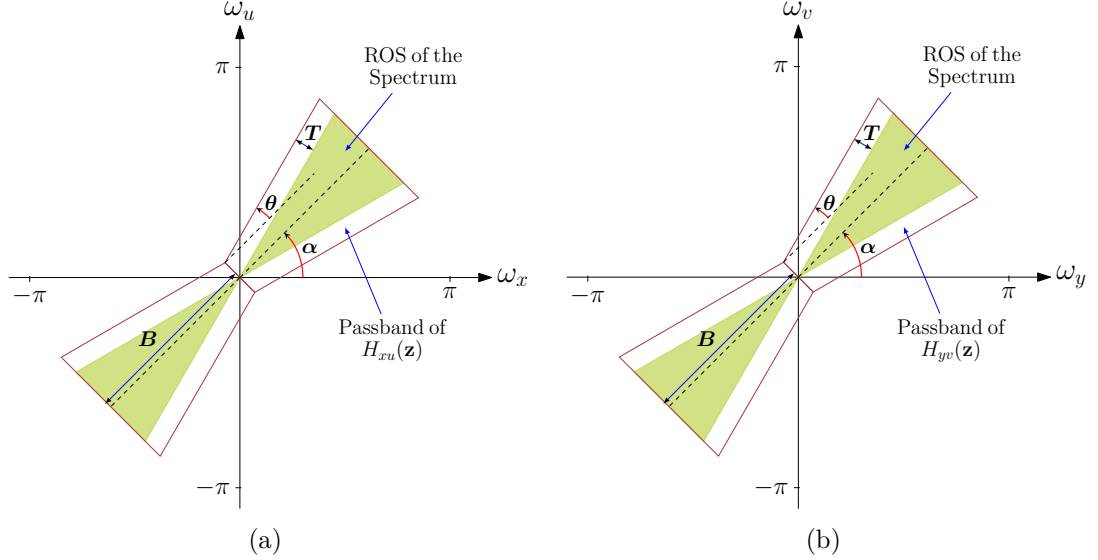


Figure 4.1: The spectral ROS of a Lambertian object and the passband of the 4-D hyperfan filter  $H(\mathbf{z})$  (a) in the  $\omega_x\omega_u$  subspace; (b) in the  $\omega_y\omega_v$  subspace.

filter that is shaped to enclose the spectral ROS of an LF, as reviewed in Section 3.3.2.

As mentioned in section 3.3.2 the ROS  $\mathcal{R}_o$  of an object is determined by modeling a Lambertian object as a collection of Lambertian point sources. Hence, spectral ROS  $\mathcal{R}_o$  was derived as [22, 68]

$$\begin{aligned}\mathcal{R}_o &= \bigcup_{z_0} \mathcal{R}_p \\ &= \bigcup_{z_0} (\mathcal{H}_{xu} \cap \mathcal{H}_{yv}).\end{aligned}\quad (4.1)$$

where,  $\mathcal{H}_{xu}$  and  $\mathcal{H}_{yv}$  represent the spectral ROS of a Lambertian point source in  $\omega_x\omega_u$  and  $\omega_y\omega_v$  subspaces respectively.

The spectral ROS  $\mathcal{R}_o$  takes a hyperfan shape in  $\boldsymbol{\omega}$  where  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_u, \omega_v) \in \mathbb{R}^4$ . Figure 4.1 depicts the two 2-D projections of the hyperfan-shaped ROS of the spectrum. Recall that, the angular width of the hyperfan depends on the depth range occupied by the Lambertian object. Hence, volumetric refocusing can be achieved with a 4-D filter having a hyperfan-shaped passband as depicted in Figure 4.1, by adjusting its angular width as necessary. This will make the objects in the depth range corresponding to the passband, appear sharply in the refocused image, whereas those having depths corresponding to the stopband, will appear blurred.



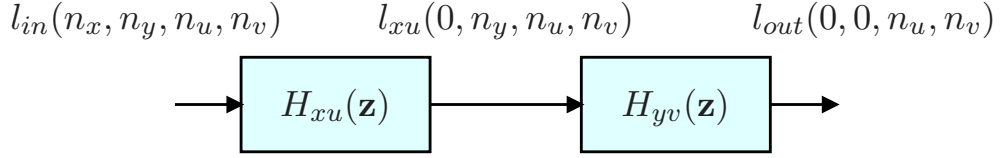


Figure 4.2: The structure of the proposed 4-D sparse FIR hyperfan filter.

### 4.3 Proposed Sparse Filter for Volumetric Refocusing

The proposed 4-D sparse FIR hyperfan filter  $H(\mathbf{z})$ ,  $[z_x, z_y, z_u, z_v]^T \in \mathbb{C}^4$  is designed as a cascade of two 4-D hyperfan filters,  $H_{xu}(\mathbf{z})$  and  $H_{yv}(\mathbf{z})$ , as illustrated in Figure 4.2. For this purpose, we employ the epipolar geometry of LFs [66] by considering  $n_x, n_u$  dimensions and  $n_y, n_v$  dimensions as separate pairs.

Utilizing the *partial separability* of the ROS, we can formulate the spectral ROS  $\mathcal{R}_o$ , as follows.

$$\mathcal{R}_o = \mathcal{B}_{xu} \cap \mathcal{B}_{yv} \quad (4.2)$$

where,

$$\mathcal{B}_{xu} = \bigcup_{z_0} \mathcal{H}_{xu} \quad (4.3a)$$

$$\mathcal{B}_{yv} = \bigcup_{z_0} \mathcal{H}_{yv}, \quad (4.3b)$$

Consequently, the passbands of  $H_{xu}(\mathbf{z})$  and  $H_{yv}(\mathbf{z})$  are selected to encompass  $\mathcal{B}_{xu}$  and  $\mathcal{B}_{yv}$  respectively. The overall passband  $H(\mathbf{z})$  fully encompasses the spectral ROS  $\mathcal{R}_o$ .

Since  $\mathcal{B}_{xu}$  is independent of  $\omega_y$  and  $\omega_v$ , and  $\mathcal{B}_{yv}$  is independent of  $\omega_x$  and  $\omega_u$ , the designs of the 4-D FIR hyperfan filters  $H_{xu}(\mathbf{z})$  and  $H_{yv}(\mathbf{z})$  reduce to the designs of 2-D FIR fan filters. In this regard, we utilize the rotation-based 2-D FIR fan filter design method proposed in [77], along with the windowing method [10, ch. 3.3]. However, for the purely fan-shaped passbands, the origin point is on the boundary between the passband and the stopband. Therefore, it gets attenuated after multiplying with a window function [77]. To mitigate this, the passbands of  $H_{xu}(\mathbf{z})$  and  $H_{yv}(\mathbf{z})$  are designed to be *bow-tie-shaped* in the  $\omega_x\omega_u$  and  $\omega_y\omega_v$  subspaces as shown in Figure 4.1. Here,  $B$  determines the length,  $\alpha$  determines the orientation and,  $\theta$  and  $T$  define the angular width of the bow-tie-shaped passbands.

Following this method, the impulse responses  $H_{xu}(\mathbf{z})$  of order  $M_x \times 0 \times M_u \times 0$  and  $H_{yv}(\mathbf{z})$  of order  $0 \times M_y \times 0 \times M_v$ , where  $M_x, M_y, M_u, M_v \in \mathbb{N}$ , are respectively

obtained as follows.

$$h_{xu}(\mathbf{n}) = \left[ h_{xu}^I(n_x, n_u) w_{xu}(n_x, n_u) \right] \delta(n_y) \delta(n_v) \quad (4.4a)$$

$$h_{yv}(\mathbf{n}) = \left[ h_{yv}^I(n_y, n_v) w_{yv}(n_y, n_v) \right] \delta(n_x) \delta(n_u), \quad (4.4b)$$

where  $\mathbf{n} = (n_x, n_y, n_u, n_v) \in \mathbb{Z}^4$ ,  $h_{xu}^I(n_x, n_u)$  and  $h_{yv}^I(n_y, n_v)$  are the ideal infinite-extent impulse responses, and  $w_{xu}(n_x, n_u)$  and  $w_{yv}(n_y, n_v)$  are 2-D separable windows of size  $(M_x + 1) \times (M_u + 1)$  and  $(M_y + 1) \times (M_v + 1)$ , respectively [10, ch. 3.3]. The derivations of  $h_{xu}^I(n_x, n_u)$  and  $h_{yv}^I(n_y, n_v)$  are elaborated in Appendix C. For an LF of size  $(N_x + 1) \times (N_y + 1) \times (N_u + 1) \times (N_v + 1)$ , where  $N_x, N_y, N_u, N_v \in \mathbb{N}$ ,  $M_x$  and  $M_y$  are set to  $N_x$  and  $N_y$ , respectively. Generally,  $M_u \ll N_u$  and  $M_v \ll N_v$ . Note that the order of the 4-D FIR hyperfan filter  $H(\mathbf{z})$  is  $M_x \times M_y \times M_u \times M_v$ . Accordingly, computational complexity of the proposed separable filter is  $O(M_x M_u + M_y M_v)$ , whereas computational complexity of the non-separable 4-D filter is  $O(M_x M_y M_u M_v)$ . Hence, computational complexity of the proposed filter is extremely low compared to that of the non-separable 4-D FIR hyperfan filter.

The two window functions  $w_{xu}(n_x, n_u)$  and  $w_{yv}(n_y, n_v)$  are chosen such that the stopband objects are only blurred rather than being completely attenuated. The 2-D separable Hamming or Kaiser windows [10, ch. 3.3], [78, ch. 6.3] are frequently employed in the cases where higher stopband attenuation is required, for instance, in designing 2-D FIR filters to attenuate noise. However, in LF volumetric refocusing, our objective is to blur the stopband objects rather than fully attenuating them. Therefore, the typically utilized Hamming and Kaiser windows are not preferable for our filter design. In planar refocusing, the refocused LF is obtained by shifting the LF in the angular dimensions  $n_u$  and  $n_v$ , and averaging in the spatial dimensions  $n_x$  and  $n_y$  [6]. Here, the shifting operation determines the focal plane of the output refocused image, whereas the averaging operation manages the blur of out-of-focus objects. Note that, averaging corresponds to lowpass filtering with a filter having a rectangular-shaped impulse response. Hence, 2-D separable *rectangular windows* are selected for  $w_{xu}(n_x, n_u)$  and  $w_{yv}(n_y, n_v)$ .

In order to obtain the sparse filter coefficients of  $H_{xu}(\mathbf{z})$  and  $H_{yv}(\mathbf{z})$ , we utilize the HT approach proposed in [74]. The sparse coefficients  $h_i^s(\mathbf{n})$  of  $H_i(\mathbf{z})$ ,  $i =$



$xu, yv$ , are obtained as,

$$h_i^s(\mathbf{n}) = \begin{cases} h_i(\mathbf{n}), & \text{if } |h_i(\mathbf{n})| \geq h_{th} \cdot \max |h_i(\mathbf{n})|, \\ 0, & \text{otherwise,} \end{cases} \quad (4.5)$$

where  $h_{th}$  is the threshold value, which is typically selected to be between 0.005 and 0.05 [74].

The volumetric refocused image is obtained for the central SAI, which we denote by  $(n_x, n_y) = (0, 0)$  where  $-N_i/2 \leq n_i \leq N_i/2$ ,  $i = x, y$ . Following this notation, the 4-D sparse FIR hyperfan filter can be denoted by the following partial-difference equations.

$$l_{xu}(0, n_y, n_u, n_v) = \underbrace{\sum_{(i_x, i_u) \in \mathcal{I}} h_{xu}^s(i_x, 0, i_u, 0)}_{(i_x, i_u) \in \mathcal{I}} \times l_{in}(-i_x, 0, n_u - i_u, 0) \quad (4.6a)$$

$$l_{out}(0, 0, n_u, n_v) = \underbrace{\sum_{(i_y, i_v) \in \mathcal{I}} h_{uv}^s(0, i_y, 0, i_v)}_{(i_y, i_v) \in \mathcal{I}} \times l_{xu}(0, -i_y, 0, n_v - i_v), \quad (4.6b)$$

Here,  $\mathcal{I}$  is the set containing the indices of the nonzero coefficients of  $H_{xu}(\mathbf{z})$  and  $H_{yv}(\mathbf{z})$ .

## 4.4 Results and Comparison

We first compare the frequency response error of the proposed 4-D sparse FIR hyperfan filter, with that of the 4-D non-sparse FIR hyperfan filter [9]. Subsequently, we compare the 4-D sparse and non-sparse FIR hyperfan filters in terms of computational complexity, and evaluate the complexity reduction of the proposed non-sparse filter. Finally, we present output images for visual comparison.

### 4.4.1 Comparison Between the 4-D Sparse and Nonsparse FIR Hyperfan Filters

The frequency response of the proposed 4-D sparse FIR hyperfan filter which is given by  $H_{xu}(\mathbf{z})$  and  $H_{yv}(\mathbf{z})$ , depends on the filter parameters ( $\alpha$ ,  $\theta$ ,  $B$  and  $T$ ), the threshold value  $h_{th}$  and the filter order. As an instance, the magnitude responses of the nonsparse and sparse  $H_{xu}(\mathbf{z})$  corresponding to  $\alpha = 50^\circ$ ,  $\theta = 20^\circ$  and  $h_{th} = 0.01$  are shown in Figs. 4.3(a) and 4.3(b), respectively. In this case, and for all the experimental results presented henceforth, we let  $B = 0.9\pi$  rad/sample,  $T = 0.08\pi$  rad/sample, and filter order= $10 \times 40$ .

Normalized-root-mean-square error (NRMSE) between the frequency responses

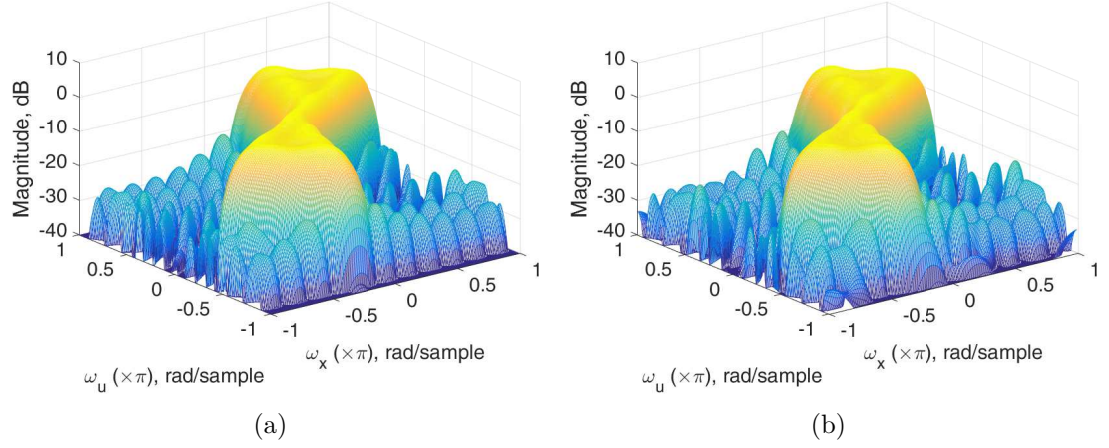


Figure 4.3: The magnitude response of  $H_{xu}(\mathbf{z})$  (a) with non-sparse coefficients; (b) with sparse coefficients.

of the sparse  $H_{xu}(\mathbf{z})$  and the non-sparse  $H_{xu}(\mathbf{z})$ , is used to quantify the deviation of the proposed 4-D sparse FIR hyperfan filter, compared to the 4-D non-sparse FIR hyperfan filter.

$$\text{NRMSE} = \frac{\|H_{xu}^s(\mathbf{k}) - H_{xu}(\mathbf{k})\|_2}{\sqrt{F_x F_u} [\max\{|H_{xu}(\mathbf{k})|\} - \min\{|H_{xu}(\mathbf{k})|\}]} \quad (4.7)$$

where,

$$\mathbf{k} = (k_x, k_u) \quad (4.8a)$$

$$H_{xu}^s(\mathbf{k}) - \text{Frequency response of the sparse filter} \quad (4.8b)$$

$$H_{xu}(\mathbf{k}) - \text{Frequency response of the non-sparse filter} \quad (4.8c)$$

$$F_x - \text{FFT length for the } n_x \text{ dimension} \quad (4.8d)$$

$$F_u - \text{FFT length for the } n_u \text{ dimension} \quad (4.8e)$$

Furthermore, number of non-zero coefficients in the 4-D sparse FIR hyperfan filter with respect to that of the non-sparse filter, is utilized as a metric to quantify the reduction in computational complexity.

Figures 4.4, 4.5 and 4.6 depict the variation of NRMSE and number of non-zero coefficients, relating to filter parameters  $\alpha$ ,  $\theta$  and threshold value  $h_{th}$ . Figure 4.4 shows the variation of NRMSE and the number of non-zero coefficients of the sparse filter compared to the non-sparse filter, with respect to  $\theta$  and  $h_{th}$  for  $\alpha = 50^\circ$ . In Figure 4.5 the same two metrics have been plotted as functions of  $\alpha$  and  $h_{th}$  while having  $\theta = 15^\circ$ , whereas in Figure 4.6 the plots have been obtained with respect to  $\theta$  and  $\alpha$ , keeping  $h_{th} = 0.01$ .

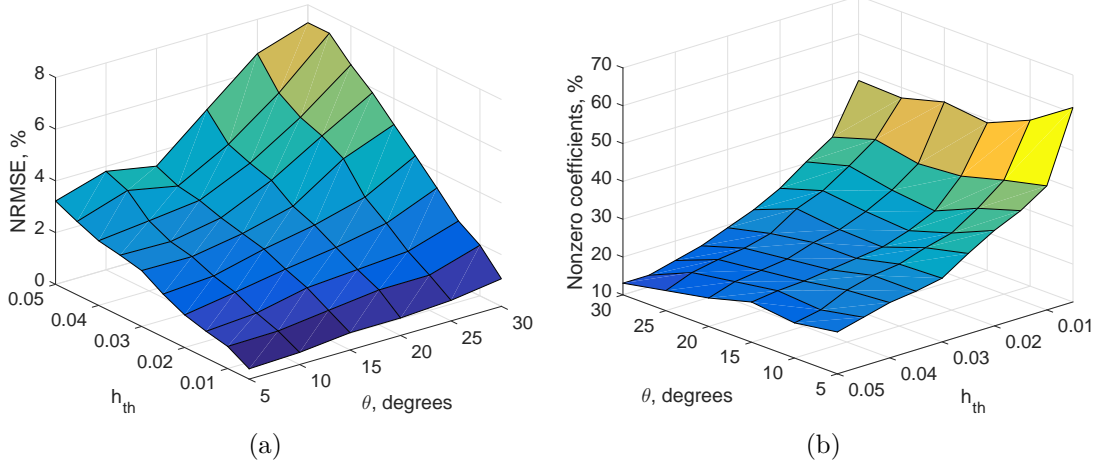


Figure 4.4: (a) NRMSE between the frequency responses of the sparse  $H_{xu}(\mathbf{z})$  and the nonsparse  $H_{xu}(\mathbf{z})$ ; (b) number of nonzero coefficients of the sparse  $H_{xu}(\mathbf{z})$  compared to the nonsparse  $H_{xu}(\mathbf{z})$ ; w.r.t.  $\theta$  and  $h_{th}$  with  $\alpha = 50^\circ$ .

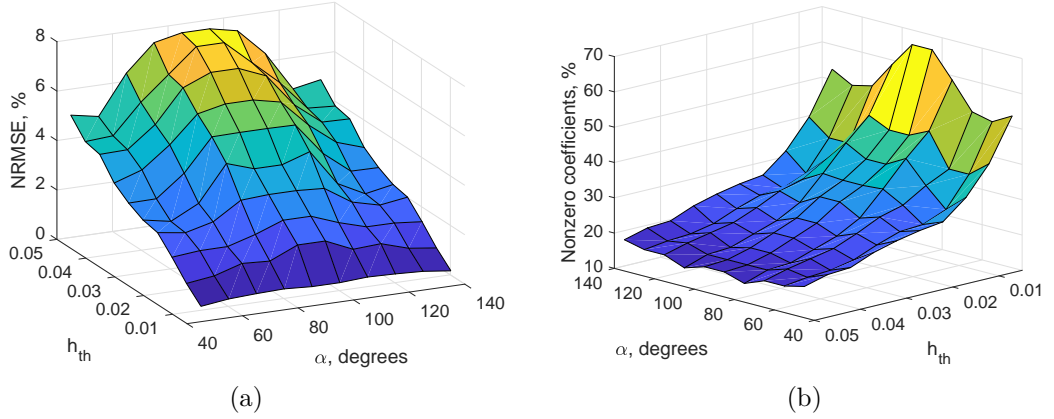


Figure 4.5: (a) NRMSE between the frequency responses of the sparse  $H_{xu}(\mathbf{z})$  and the nonsparse  $H_{xu}(\mathbf{z})$ ; (b) number of nonzero coefficients of the sparse  $H_{xu}(\mathbf{z})$  compared to the nonsparse  $H_{xu}(\mathbf{z})$ ; w.r.t.  $\alpha$  and  $h_{th}$  with  $\theta = 15^\circ$ .

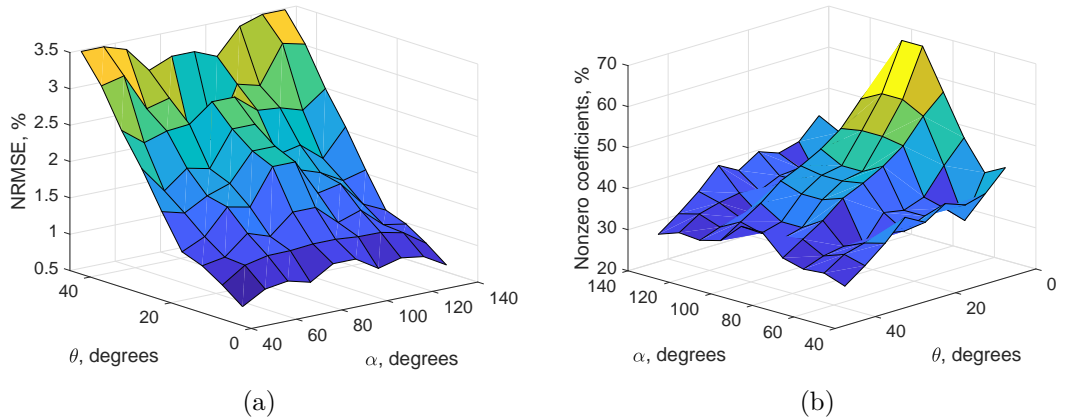


Figure 4.6: (a) NRMSE between the frequency responses of the sparse  $H_{xu}(\mathbf{z})$  and the nonsparse  $H_{xu}(\mathbf{z})$ ; (b) number of nonzero coefficients of the sparse  $H_{xu}(\mathbf{z})$  compared to the nonsparse  $H_{xu}(\mathbf{z})$ ; w.r.t.  $\alpha$  and  $\theta$  with  $h_{th} = 0.01$ .

Table 4.1: The mean and standard deviation of NRMSE and number of non-zero coefficients for the instances considered in figures 4.4, 4.5 and 4.6.

Figure	NRMSE		Nonzero coeff.	
	mean	std dev	mean	std dev
4.4	2.83%	1.71%	28.19%	10.39%
4.5	3.63%	1.93%	28.81%	11.73%
4.6	1.91%	0.8%	39.74%	8.62%

Table 4.2: The means and the standard deviations of the SSIM indices between the volumetric refocused images obtained with the proposed sparse and the nonsparse hyperfan filters [9].

LF	MPT	Books	Flowers	SV1	GG
Mean	0.986	0.991	0.989	0.989	0.989
Standard deviation	0.019	0.012	0.014	0.017	0.015

MPT - Mirabelle Prune Tree, SP1 - Sophie & Vincent 1, and GG - Gravel Garden

The mean and standard deviation of NRMSE and the number of nonzero coefficients corresponding to above three cases, are given in Table 4.1. By observing the tabulated values it is evident that the deviation of  $H_{xu}^s(\mathbf{z})$  compared to  $H_{xu}(\mathbf{z})$  is *negligible*. Furthermore,  $H_{yv}(\mathbf{z})$  and  $H_{yv}^s(\mathbf{z})$  possess the same characteristics as  $H_{xu}(\mathbf{z})$  and  $H_{xu}^s(\mathbf{z})$ , respectively.

Moreover, sparse  $H_{xu}^s(\mathbf{z})$  provides *approximately 72% mean reduction in computational complexity*, since zero-valued coefficients do not require arithmetic operations. As an example, consider an LF of size  $11 \times 11 \times 512 \times 512$  and filter order of  $10 \times 10 \times 40 \times 40$ . The non-sparse filter requires 2172 multiplications and 5400 additions, whereas the sparse filter requires only 768 multiplications and 1512 additions, considering direct-form realizations and symmetry of the filter coefficients. Additionally, the frequency domain implementation of the non-sparse filter [9], requires 3993 multiplications and 19239 additions, with the split-radix algorithm for  $n_u$  and  $n_v$  dimensions, and Winograd algorithm for  $n_x$  and  $n_y$  dimensions [79, ch. 3].

#### 4.4.2 Performance of the 4-D Sparse FIR Hyperfan Filter in Volumetric Refocusing





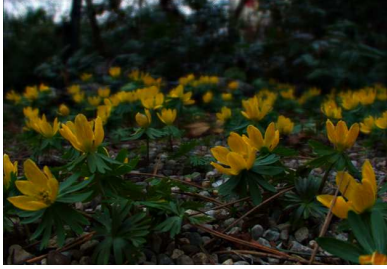
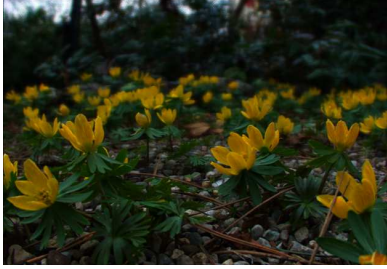

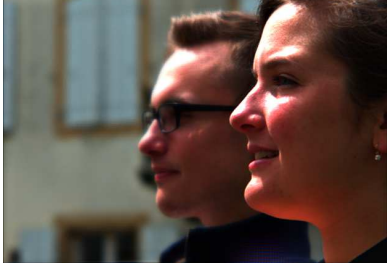


The performance of the proposed sparse filter in volumetric refocusing, is compared with that of the 4-D non-sparse FIR hyperfan filter using multiple LFs from the EPFL LF dataset [62, 70]. Both sparse and nonsparse filters are designed with filter order  $10 \times 10 \times 40 \times 40$  and parameters  $\alpha = 50^\circ$ ,  $B = 0.9\pi$

rad/sample, and  $T = 0.08\pi$ .  $\theta$  and  $h_{th}$  are varied in the range  $5^\circ$  to  $30^\circ$  with the steps of  $5^\circ$  and 0.005 to 0.05 with the steps of 0.005. The mean and standard deviations of the SSIM indices, between the outputs of the proposed sparse filter and the nonsparse filter, are presented in Table 4.2. From these results we can conclude that, the distortion caused by the sparsity of the filter coefficients of the proposed filter, is *negligible*. The volumetric refocused images of multiple LFs, obtained for  $\alpha = 45^\circ$  and  $\theta = 35^\circ$  are given in Table 4.3 for visual comparison. Through visual inspection it is evident that, volumetric refocused images of the proposed sparse filter have *no visual difference* from those of the non-sparse filter. Table 4.4 shows the outputs of the sparse volumetric refocusing filter for two  $\alpha$  values. As stated before,  $\alpha$  corresponds to the angle of orientation of the 4-D FIR hyperfan filters  $H_{xu}(\mathbf{z})$  and  $H_{yv}(\mathbf{z})$ . Recall that, depth  $z_0$  of a Lambertian point source corresponds to the slope  $m$  of a hyperplane through the origin of the 4-D frequency domain. This relationship is characterized in (3.8) as  $\alpha = \tan^{-1}\left(\frac{z_0}{D}\right)$  where,  $m = \tan \alpha$  and  $D$  is the constant distance between the parallel planes in the two-plane parameterization. Hence,  $\alpha = 60^\circ$  corresponds to a closer depth whereas  $\alpha = 105^\circ$  corresponds to a more distant depth. Furthermore, the range of depths in focus is kept similar for both sparse filter and non-sparse filter, by keeping  $\theta$  and  $T$  constant. The output images in Table 4.4 clearly contrasts refocusing on different depths.

The experiments presented in this chapter so far, have been conducted assuming that the input LF is already denoised. Table 4.5 explores how the proposed sparse filter, its non-sparse counterpart and the hyperfan filter [12] behave in case the input LF is noisy. Each output image obtained for a noisy LF, is compared with the central SAI of the noise-free LF. From the PSNR and SSIM values tabulated in Table 4.5, it is evident that the proposed filter is only suitable for generating refocused views from previously denoised LFs. Compared to this, the hyperfan filter denoises the LF as an inherent part of refocusing.

The execution time for the proposed sparse cascaded filter, the non-sparse cascaded filter, and the 4-D hyperfan filter [12] was measured using the MATLAB implementations on the same Windows 10 system powered by an Intel i-7 processor. The average time recorded for the proposed sparse filter and its non-sparse counterpart is 13 seconds whereas the average time consumption of the 4-D hyperfan is 15 seconds. Although theoretically the number of computations has drastically reduced, it has not been reflected in the execution time. This is because the FFT algorithms utilized in the implementation are not optimized for sparsity. However, in hardware implementations using field programmable gate





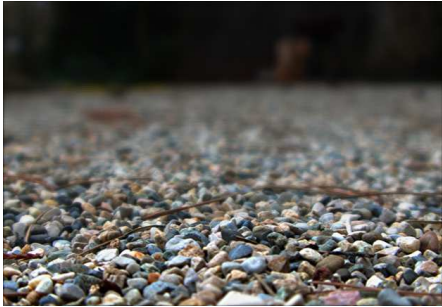


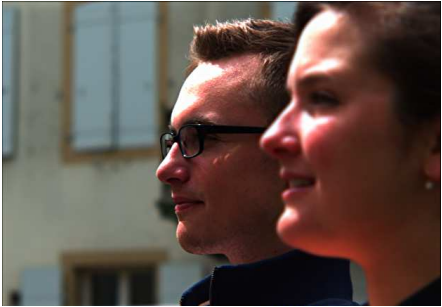
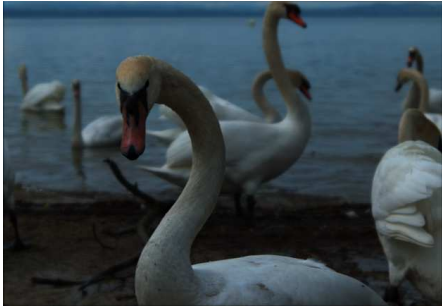

Table 4.3: Volumetric refocused images obtained with the proposed sparse filter and the non-sparse filter

LF	Sparse filter	Non-sparse filter	SSIM
MPT			0.9714
Books			0.9915
Flowers			0.9882
SV1			0.9897
GG			0.9948

MPT - Mirabelle Prune Tree, SV1 - Sophie & Vincent 1, and GG - Gravel Garden



Table 4.4: Volumetric refocused images obtained with the proposed sparse filter for  $\alpha = 60^\circ$  and  $\alpha = 105^\circ$

LF	$\alpha = 60^\circ$	$\alpha = 105^\circ$
Books		
Flowers		
GG		
SV1		
Swans 1		

GG - Gravel Garden and SV1 - Sophie & Vincent 1

Table 4.5: Output comparison for color noisy LFs with noise variance  $\sigma^2=0.02$ 

LF	Noisy LF		Non-sparse		Sparse		Hyperfan	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Rolex Learning Center	17.86	0.2284	19.56	0.2922	19.10	0.2973	25.26	0.7338
Houses & Lake	18.38	0.4454	20.18	0.4941	19.33	0.4937	27.31	0.7612
Reeds	17.84	0.2798	19.73	0.3494	19.15	0.3554	28.56	0.7593
Graffiti	17.58	0.3305	19.33	0.3858	18.48	0.3986	28.04	0.7889
Diplodocus	17.33	0.4944	19.14	0.5794	18.43	0.5988	28.77	0.9257

arrays (FPGAs) or application specific integrated circuits, the proposed sparse filter leads to significant resource saving compared to the non-sparse filter.

The proposed sparse FIR hyperfan filter provides a significant reduction in computational complexity with negligible distortion. Although the HT approach employed here is a fairly simple technique, it has produced an impressive outcome in this case. However, the proposed filter has a limitation in that it is not suitable for denoising as an inherent part of refocusing, whereas the 4-D hyperfan [12] can be used for both tasks.



## Chapter 5

### CONCLUSIONS AND FUTURE WORK

#### 5.1 Conclusions

In this thesis, we propose two novel *low-complexity* LF processing algorithms focusing on LF denoising and volumetric refocusing. To this end, the ROS of the LF of a Lambertian object is analyzed, and it is shown that the spectral ROS of a Lambertian object takes a hyperfan shape in the 4-D frequency domain. In the spectral analysis, it is first demonstrated that the spectral ROS of a Lambertian point source is a plane through the origin in the 4-D frequency domain. The planar ROS lies at the intersection of two 3-D hyperplanes, whose orientation depends on the depth of the point source. Extending this, it is also demonstrated that the angular width of the hyperfan-shaped ROS depends on the range of depths occupied by the Lambertian object.

Utilizing the sparsity of the ROS of an LF, it is shown that noise can be attenuated by *selective filtering*. Here, we only filter the regions in the 4-D frequency domain where LF signal is present. Observing the spectral ROS of LFs from the EPFL LF dataset [62], we employ filters with parallelogram and circular shaped passbands for 2-D filtering in the angular (u,v) dimensions. Furthermore, using a *mixed-domain* approach, LF is processed in the spatial domain for spatial (x,y) dimensions and in the frequency domain for the angular (u,v) dimensions. By theoretical analysis it is shown that the computational and memory complexity of the proposed algorithm is less than those of the available linear LF denoising methods. The experimental results indicate that the proposed filter predominantly outperforms the competing linear methods, in terms of PSNR and SSIM. However, the non-linear LF denoising methods outperform all the linear methods including the proposed algorithm. Furthermore, the proposed filter executes faster than both linear and non-linear LF denoising methods.

Utilizing the *partial separability* of the ROS of an LF, we employ a cascade of two 4-D hyperfan filters [9] for volumetric refocusing of LFs. The designs of the 4-D FIR hyperfan filters reduce to the designs of 2-D FIR fan filters since the

four dimensions are independent of each other. The sparse coefficients of the two fan filters are obtained by utilizing the HT approach. The experimental results demonstrate that the distortion caused by the sparsity of the proposed filter is negligible, while providing significant reduction in computational complexity.

With the growing interest in robotics community to employ LFs for visual odometry [25–28], there is an increasing requirement of *real-time* algorithms that are simple enough to run on smaller processing units employed in autonomous robotics. Similarly, the prospect of employing LF cameras in mobile phones, also has created a requirement for *real-time* LF processing algorithms, that can run on the limited processing power and limited memory in mobile phones. Hence, the algorithms presented in this work have a considerable potential in catering to this requirement.

## 5.2 Future Work

True real-time processing of the proposed denoising method can be achieved with a FPGA implementation which can process tens of LF frames per a second. Furthermore, such an FPGA implementation would drastically reduce the memory complexity of the filter as well. Another direction to work on, is using low-complexity infinite-extent impulse response (IIR) planar filters in place of FIR planar filters. Additionally, we can extend the proposed denoising algorithm to 5-D light field video (LFV) denoising as well [80].

## References

- [1] E. H. Adelson and J. R. Bergen, “The plenoptic function and the elements of early vision,” in *Computation Models of Visual Processing*, M. Landy and J. A. Movshon, Eds. Cambridge, MA: MIT Press, 1991, pp. 3–20.
- [2] M. Levoy and P. Hanrahan, “Light field rendering,” in *Proc. Annu. Conf. Comput. Graph. (SIGGRAPH)*, 1996, pp. 31–42.
- [3] D. Dansereau and L. Bruton, “A 4D frequency-planar IIR filter and its application to light field processing,” in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, 2003, pp. IV–476–IV–479.
- [4] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, “Light field image processing: An overview,” *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 926–954, Oct. 2017.
- [5] A. Madanayake, C. Wijenayake, D. G. Dansereau, T. K. Gunaratne, L. T. Bruton, and S. B. Williams, “Multidimensional (MD) circuits and systems for emerging applications including cognitive radio, radio astronomy, robot vision and imaging,” *IEEE Circuits Syst. Mag.*, vol. 13, no. 1, pp. 10–43, 2013.
- [6] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” Stanford Univ., Stanford, CA, Tech. Rep. CTSR 2005-02, 2005.
- [7] O. Pizarro, S. B. Williams, M. V. Jakuba, M. Johnson-Roberson, I. Mahon, M. Bryson, D. Steinberg, A. Friedman, D. Dansereau, N. Nourani-Vatani *et al.*, “Benthic monitoring with robotic platforms—the experience of australia,” in *Proc. IEEE Int. Symp. Underwater Technology*. IEEE, 2013, pp. 1–10.
- [8] D. G. Dansereau, “Plenoptic signal processing for robust vision in field robotics,” Ph.D. dissertation, 2014.

- [9] D. G. Dansereau, O. Pizarro, and S. B. Williams, “Linear volumetric focus for light field cameras,” *ACM Trans. Graph.*, vol. 34, no. 2, pp. 15:1–15:20, Feb. 2015.
- [10] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [11] S. U. Premaratne, C. U. Edussooriya, C. Wijenayake, L. T. Bruton, and P. Agathoklis, “A 4-D sparse FIR hyperfan filter for volumetric refocusing of light fields by hard thresholding,” in *Proc. IEEE Int. Conf. Digital Signal Process.* IEEE, 2018, pp. 1–5.
- [12] D. G. Dansereau, D. L. Bongiorno, O. Pizarro, and S. B. Williams, “Light field image denoising using a linear 4D frequency-hyperfan all-in-focus filter,” in *Proc. SPIE Comput. Imag. XI*, vol. 8657, 2013, pp. 86 570P–1–86 570P–14.
- [13] L. McMillan and G. Bishop, “Plenoptic modeling: An image-based rendering system,” in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1995, pp. 39–46.
- [14] C. Zhang and T. Chen, “A survey on image-based rendering—representation, sampling and compression,” *Signal Process.: Image Commun.*, vol. 19, no. 1, pp. 1–28, Jan. 2004.
- [15] H.-Y. Shum, S. B. Kang, and S.-C. Chan, “Survey of image-based representations and compression techniques,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.
- [16] C. U. S. Edussooriya, D. G. Dansereau, L. T. Bruton, and P. Agathoklis, “Five-dimensional depth-velocity filtering for enhancing moving objects in light field videos,” *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2151–2163, Apr. 2015.
- [17] E. H. Adelson and J. Y. Wang, “Single lens stereo with a plenoptic camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 99–106, 1992.
- [18] R. Ng, “Digital light field photography,” Ph.D. dissertation, Stanford University, CA, USA, 2006.
- [19] S. B. Kang, “Survey of image-based rendering techniques,” in *Videometrics VI*, vol. 3641. International Society for Optics and Photonics, 1998, pp. 2–17.

- [20] M. Levoy, “Light fields and computational imaging,” *Computer*, vol. 39, no. 8, pp. 46–55, Aug 2006.
- [21] A. Isaksen, L. McMillan, and S. J. Gortler, “Dynamically reparameterized light fields,” in *Proc. Annu. Conf. Comput. Graph. (SIGGRAPH)*, 2000, pp. 297–306.
- [22] D. Dansereau and L. T. Bruton, “A 4-D dual-fan filter bank for depth filtering in light fields,” *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 542–549, Feb. 2007.
- [23] N. Liyanage, C. Wijenayake, C. U. S. Edussooriya, A. Madanayake, P. Agathoklis, E. Ambikairajah, and L. Bruton, “Low-complexity 4-D IIR filters for multi-depth filtering and occlusion suppression in light fields,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2018, pp. 1–5.
- [24] J. Yu, “A light-field journey to virtual reality,” *IEEE MultiMedia*, vol. 24, no. 2, pp. 104–112, 2017.
- [25] D. G. Dansereau, I. Mahon, O. Pizarro, and S. B. Williams, “Plenoptic flow: Closed-form visual odometry for light field cameras,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2011, pp. 4455–4462.
- [26] F. Dong, S.-H. Ieng, X. Savatier, R. Etienne-Cummings, and R. Benosman, “Plenoptic cameras in real-time robotics,” *The International Journal of Robotics Research*, vol. 32, no. 2, pp. 206–217, 2013.
- [27] N. Zeller, F. Quint, and U. Stilla, “From the calibration of a light-field camera to direct plenoptic odometry,” *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1004–1019, 2017.
- [28] M. Lingenauber, K. H. Strobl, N. W. Oumer, and S. Kriegel, “Benefits of plenoptic cameras for robot vision during close range on-orbit servicing maneuvers,” in *IEEE Conf. Aerosp.* IEEE, 2017, pp. 1–18.
- [29] S. Hasirlioglu, M. Karthik, A. Riener, and I. Doric, “Potential of plenoptic cameras in the field of automotive safety,” in *First International Conference on Intelligent Transport Systems*. Springer, 2017, pp. 164–173.
- [30] A. Sepas-Moghaddam, F. Pereira, and P. L. Correia, “Ear recognition in a light field imaging framework: A new perspective,” *IET Biometrics*, vol. 7, no. 3, pp. 224–231, 2018.

- [31] S. Shi, J. Wang, J. Ding, Z. Zhao, and T. New, “Parametric study on light field volumetric particle image velocimetry,” *Flow Measurement and Instrumentation*, vol. 49, pp. 70–88, 2016.
- [32] A. Sepas-Moghaddam, F. Pereira, and P. L. Correia, “Light field-based face presentation attack detection: reviewing, benchmarking and one step further,” *IEEE Trans. Inf. Forens. Security*, vol. 13, no. 7, pp. 1696–1709, 2018.
- [33] X. Zhou, “Sampling models in light fields,” Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, 2016.
- [34] Lytro Inc. [Online]. Available: <https://www.lytro.com/>
- [35] ——. [Online]. Available: <https://www.lytro.com/illum/>
- [36] Raytrix GmbH. [Online]. Available: <https://www.raytrix.de/>
- [37] M. Levoy. (2002) The stanford multi-camera array. [Online]. Available: <http://graphics.stanford.edu/projects/array/>
- [38] 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,” in *Proc. Annu. Conf. Comput. Graph. (SIGGRAPH)*, 2005, pp. 765–776.
- [39] M. Levoy. (2005) Stanford light field microscope project. [Online]. Available: <http://graphics.stanford.edu/projects/lfmicroscope/>
- [40] ——. (2004) Stanford spherical gantry. [Online]. Available: <https://graphics.stanford.edu/projects/gantry/>
- [41] K. Venkataraman, D. Lelescu, J. Duparré, A. McMahon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar, “PiCam: An ultra-thin high performance monolithic camera array,” *ACM Trans. Graph.*, vol. 32, no. 6, pp. 166:1–166:13, Nov. 2013.
- [42] A. J. Theuwissen, “CMOS image sensors: State-of-the-art,” *Solid-State Electronics*, vol. 52, no. 9, pp. 1401–1406, 2008.
- [43] H. Tian, “Noise analysis in CMOS image sensors,” Ph.D. dissertation, Department of Applied Physics, Stanford University, CA,USA, 2000.

- [44] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. CRC press, 2016.
- [45] P. Jain and V. Tyagi, “A survey of edge-preserving image denoising methods,” *Information Systems Frontiers*, vol. 18, no. 1, pp. 159–170, 2016.
- [46] 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 Trans. Cybern.*, vol. 44, no. 7, pp. 1001–1013, 2014.
- [47] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [48] A. Beck and M. Teboulle, “Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems,” *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009.
- [49] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [50] K. Dabov, A. Foi, and K. Egiazarian, “Video denoising by sparse 3D transform-domain collaborative filtering,” in *2007 15th European Signal Processing Conference*, Sep. 2007, pp. 145–149.
- [51] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, “Video denoising, de-blocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms,” *IEEE Trans. Image Process*, vol. 21, no. 9, pp. 3952–3966, 2012.
- [52] Z. Li, H. Baker, and R. Bajcsy, “Joint image denoising using light-field data,” in *IEEE Int. Conf. Multimedia and Expo Workshops*. IEEE, 2013, pp. 1–6.
- [53] A. Sepas-Moghaddam, P. L. Correia, and F. Pereira, “Light field denoising: exploiting the redundancy of an epipolar sequence representation,” in *Proc. IEEE 3DTV Conf.* IEEE, 2016, pp. 1–4.
- [54] M. Alain and A. Smolic, “Light field denoising by sparse 5D transform domain collaborative filtering,” in *IEEE Int. Workshop on Multimedia Signal Process. (MMSP)*. IEEE, 2017, pp. 1–6.

- [55] J. Chen, J. Hou, and L.-P. Chau, “Light field denoising via anisotropic parallax analysis in a CNN framework,” *arXiv preprint arXiv:1805.12358*, 2018.
- [56] P. Allain, L. Guillo, and C. Guillemot, “Light field denoising using 4d anisotropic diffusion,” in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2019, pp. 1692–1696.
- [57] A. Bajpayee, A. H. Techet, and H. Singh, “Real-time light field processing for autonomous robotics,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.* IEEE, 2018, pp. 4218–4225.
- [58] N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy, “Synthetic depth-of-field with a single-camera mobile phone,” *ACM Trans. Graph.*, vol. 37, no. 4, p. 64, 2018.
- [59] L. Wang, X. Shen, J. Zhang, O. Wang, Z. Lin, C.-Y. Hsieh, S. Kong, and H. Lu, “Deeplens: shallow depth of field from a single image,” *arXiv preprint arXiv:1810.08100*, 2018.
- [60] R. Ng, “Fourier slice photography,” in *Proc. Annu. Conf. Comput. Graph. (SIGGRAPH)*, 2005, pp. 735–744.
- [61] J. Fiss, B. Curless, and R. Szeliski, “Refocusing plenoptic images using depth-adaptive splatting,” in *Proc. Int. Conf. Comput. Photogr.*, 2014, pp. 1–9.
- [62] M. Rerabek and T. Ebrahimi. (2016) New light field image dataset. [Online]. Available: <http://mmspg.epfl.ch/EPFL-light-field-image-dataset>
- [63] R. Szeliski, *Computer Vision: Algorithms and Applications*. London: Springer, 2011.
- [64] K. J. Dana, B. Van Ginneken, S. K. Nayar, and J. J. Koenderink, “Reflectance and texture of real-world surfaces,” *ACM Trans. Graph.*, vol. 18, no. 1, pp. 1–34, 1999.
- [65] D. G. Dansereau, “4D light field processing and its application to computer vision,” Master’s thesis, AB, Canada, 2003.



- [66] R. C. Bolles, H. H. Baker, and D. H. Marimont, “Epipolar-plane image analysis: An approach to determining structure from motion,” *Int. J. Comput. Vision*, vol. 1, no. 1, pp. 7–55, 1987.
- [67] C. U. S. Edussooriya, “Low-complexity multi-dimensional filters for plenoptic signal processing,” Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, 2015.
- [68] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, “Plenoptic sampling,” in *Proc. Annu. Conf. Comput. Graph. (SIGGRAPH)*, 2000, pp. 307–318.
- [69] A. Levin, S. W. Hasinoff, P. Green, F. Durand, and W. T. Freeman, “4D frequency analysis of computational cameras for depth of field extension,” in *Proc. Annu. Conf. Comput. Graph. (SIGGRAPH)*, 2009, pp. 97:1–97:14.
- [70] M. Rerabek and T. Ebrahimi, “New light field image dataset,” in *Proc. 8th Int. Conf. Quality Multimedia Experience*, 2016, pp. 1–2. [Online]. Available: <http://mmspg.epfl.ch/EPFL-light-field-image-dataset>
- [71] Computer Graphics Laboratory, Stanford University. (2008) The (New) Stanford Light Field Archive. [Online]. Available: <http://lightfield.stanford.edu/lfs.html>
- [72] Y. Liu, N. Qi, Z. Cheng, D. Liu, Q. Ling, and Z. Xiong, “Tensor-based light field denoising by integrating super-resolution,” in *IEEE Int. Conf. Image Process. (ICIP)*. IEEE, 2018, pp. 3209–3213.
- [73] S. Fujita, K. Takahashi, and T. Fujii, “How should we handle 4d light fields with cnns?” in *IEEE Int. Conf. Image Process (ICIP)*, Oct 2018, pp. 2600–2604.
- [74] L. Khademi and L. T. Bruton, “Reducing the computational complexity of narrowband 2D fan filters using shaped 2D window functions,” in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, 2003, pp. III–702–III–705.
- [75] R. T. Wijesekara, C. U. S. Edussooriya, L. T. Bruton, and P. Agathoklis, “A low-complexity 2-D spatially-interpolated FIR trapezoidal filter for enhancing broadband plane waves,” in *Proc. IEEE Int. Workshop Multidim. Syst.*, 2017, pp. 1–6.
- [76] —, “A 3-D sparse FIR frustum filter for enhancing broadband plane waves,” *IEEE Trans. Circuits Syst. II*, pp. 1–5, 2018.

- [77] S.-C. Pei and S.-B. Jaw, “Two-dimensional general fan-type FIR digital filter design,” *Signal Process.*, vol. 37, no. 2, pp. 265–274, May 1994.
- [78] W.-S. Lu and A. Antoniou, *Two-Dimensional Digital Filters*. NY: Marcel Dekker, 1992.
- [79] R. E. Blahut, *Fast Algorithms for Signal Processing*. NY: Cambridge University Press, 2010.
- [80] C. U. S. Edussooriya, L. T. Bruton, and P. Agathoklis, “A novel 5-D depth-velocity filter for enhancing noisy light field videos,” *Multidimensional Systems and Signal Processing*, vol. 28, no. 1, pp. 353–369, 2017.
- [81] J. W. Woods, *Multidimensional Signal, Image, and Video Processing and Coding*. Elsevier, 2006.

## Appendix A

### Derivation of the Ideal Infinite-Extent Impulse Response $g_{uv}^I(n_u, n_v)$ of Parallelogram Filter

The frequency response  $H_{uv}(e^{j\omega})$  of the parallelogram filter, depicted in Figure 3.10(a), can be expressed as follows.

$$H_{uv}(e^{j\omega}) = \begin{cases} 1, & \omega_{u_{min}} \leq \omega_u \leq \omega_{u_{max}}, \\ & -C_v \leq \omega_v \leq C_v, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.1})$$

where,

$$\begin{aligned} \omega_{u_{min}} &= \omega_v \cot \alpha - b_u \\ \omega_{u_{max}} &= \omega_v \cot \alpha + b_u \end{aligned}$$

The infinite extent impulse response  $h_{uv}^I(\mathbf{n})$  of the parallelogram filter where  $\mathbf{n} = (n_x, n_y, n_u, n_v)$ , is acquired by obtaining the 4-D inverse Fourier transform of its frequency response  $H_{uv}(e^{j\omega})$  [16].

$$h_{uv}^I(\mathbf{n}) = g_{uv}^I(n_u, n_v) \delta(n_x) \delta(n_y) \quad (\text{A.2})$$

where,

$$g_{uv}^I(n_u, n_v) = \begin{cases} \frac{C_v B \csc(\alpha)}{2\pi^2}, & n_u = 0, n_v = 0, \\ \frac{B \csc(\alpha) \sin(n_v C_v)}{2\pi^2 n_v}, & n_u = 0, n_v \neq 0, \\ \frac{C_v \sin\left(n_u \frac{B}{2} \csc \alpha\right)}{\pi^2 n_u}, & n_u \neq 0, n_u \cot \alpha + n_v = 0, \\ \sin(n_u b_u) \times \frac{\sin[(n_u \cot \alpha + n_v) C_v]}{\pi^2 n_u (n_u \cot \alpha + n_v)}, & \text{otherwise.} \end{cases} \quad (\text{A.3})$$

## Appendix B

### Derivation of the Ideal Infinite-Extent Impulse Response $k_{uv}^I(n_u, n_v)$ of Circular Filter

The frequency response  $H_{uv}(e^{j\omega})$  of the circular filter, depicted in Figure 3.10(b), can be expressed as follows.

$$H_{uv}(e^{j\omega}) = \begin{cases} 1, & \omega_u^2 + \omega_v^2 \leq \omega_c^2, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{B.1})$$

The infinite extent impulse response  $k_{uv}^I(\mathbf{n})$  of the parallelogram filter where  $\mathbf{n} = (n_x, n_y, n_u, n_v)$ , is acquired by obtaining the 4-D inverse Fourier transform of its frequency response  $H_{uv}(e^{j\omega})$  [81, ch. 1.2].

$$k_{uv}^I(n_u, n_v) = \begin{cases} \frac{\omega_c}{2\pi\sqrt{n_u^2+n_v^2}} J_1(\omega_c\sqrt{n_u^2+n_v^2}), & n_u \neq 0, n_v \neq 0, \\ \frac{\omega_c^2}{2}, & n_u = 0, n_v = 0, \end{cases} \quad (\text{B.2})$$

where  $J_1$  is first order Bessel function of the first kind.

## Appendix C

### Derivation of the Bow-tie Shaped Passband

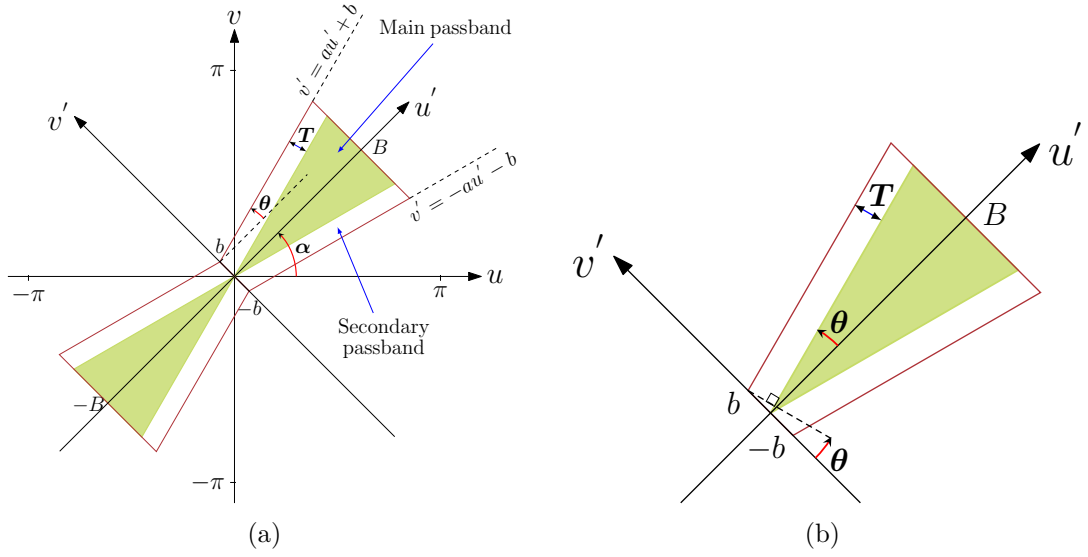


Figure C.1: (a) Bowtie-shaped passband (b) Enlarged portion of (a)

Here, we review the derivation of the bow-tie shaped passband. The generalized fan-shaped filter is shown in Figure C.1(a). A secondary passband is added around the main passband in order to prevent the attenuation at the zero-frequency response. Using the line equation  $v' = au' + b$ , we can obtain the following relationships for  $a$  and  $b$  from Figure C.1(b).

$$a = \tan \theta, \quad (\text{C.1a})$$

$$b = \frac{T}{\cos \theta}, \quad (\text{C.1b})$$

where,  $T$  is the width of the secondary passband.

The ideal impulse response  $h(m, n)$  can be written as [77],

$$h(m, n) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H(e^{ju}, e^{jv}) e^{j(mu+nv)} dudv. \quad (\text{C.2})$$

For convenience in integration, we use rotated coordinate system  $(u', v')$  instead of the original  $(u, v)$  coordinate system. The relationship between the two coordinate systems is given by a rotation operation as follows.

$$(u', v') = (u, v) \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \quad (\text{C.3})$$

Using the Jacobian,

$$\begin{aligned} dudv &= \begin{vmatrix} \frac{\partial u}{\partial u'} & \frac{\partial v}{\partial u'} \\ \frac{\partial u}{\partial v'} & \frac{\partial v}{\partial v'} \end{vmatrix} du' dv' \\ &= \begin{vmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{vmatrix} du' dv' \\ &= du' dv' \end{aligned} \quad (\text{C.4})$$

Then C.2 can be rewritten as,

$$\begin{aligned} h(m, n) &= \frac{2}{4\pi^2} \int_0^B \int_{-(au'+b)}^{au'+b} \cos[mu' \cos \alpha - mv' \sin \alpha + nu' \sin \alpha + nv' \cos \alpha] du' dv', \\ &= \frac{2}{4\pi^2} \int_0^B \int_{-(au'+b)}^{au'+b} \cos(pu' + qv') du' dv', \\ &= g(p, q), \end{aligned} \quad (\text{C.5})$$

where,

$$(p, q) = (m, n) \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \quad (\text{C.6})$$

Closed form solutions of C.5 can be specified for special conditions as follows [77].

$$\begin{aligned} g(p, q) &= \frac{1}{4\pi^2 q} \left[ \frac{\cos(qb) - \cos(pB + qaB + qb)}{p + qa} - \frac{\cos(qb) - \cos(pB - qaB - qb)}{p - qa} \right] \\ &\quad \text{for } q \neq 0, p + qa \neq 0, p - qa \neq 0, \end{aligned} \quad (\text{C.7})$$

$$g(0, 0) = \frac{B}{4\pi^2} (aB + 2b) \quad \text{for } p = q = 0, \quad (\text{C.8})$$

$$g(p, 0) = \frac{1}{2\pi^2 p} \left[ (aB + b) \sin(pB) + \frac{a(\cos(pB) - 1)}{p} \right] \quad \text{for } q = 0, \quad (\text{C.9})$$

$$g(p, q) = \frac{1}{4\pi^2 q} \left[ B \sin(qb) - \frac{\cos(qb) - \cos(pB - qaB - qb)}{p - qa} \right] \quad \text{for } p + qa = 0, \quad (\text{C.10})$$

$$g(p, q) = \frac{1}{4\pi^2 q} \left[ B \sin(qb) + \frac{\cos(qb) - \cos(pB + qaB + qb)}{p + qa} \right] \quad \text{for } p - qa = 0. \quad (\text{C.11})$$

## Appendix D

### Supplementary Results for the Proposed Low-Complexity Denoising Algorithm

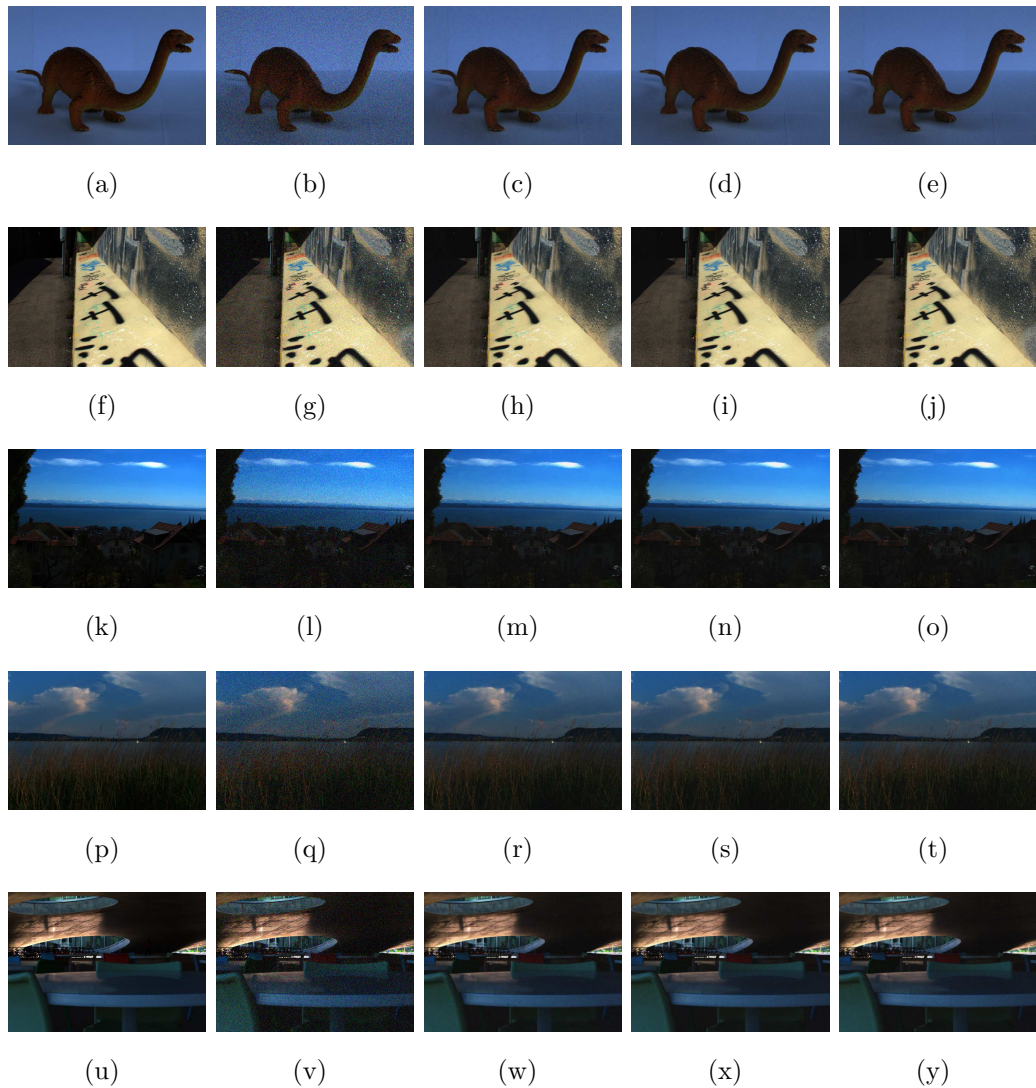


Figure D.1: SAIs of LFs from EPFL dataset for denoising with  $\sigma = 0.1$ . From left, first column: original image, second column: noisy image, third column: output of proposed method, fourth column: output of 4-D hyperfan filter, fifth column: output of 4-D planar filter, (a)-(e) Diplodocus (f)-(j) Graffiti (k)-(o) Houses & Lake (p)-(t) Reeds (u)-(y) Rolex Learning Center.



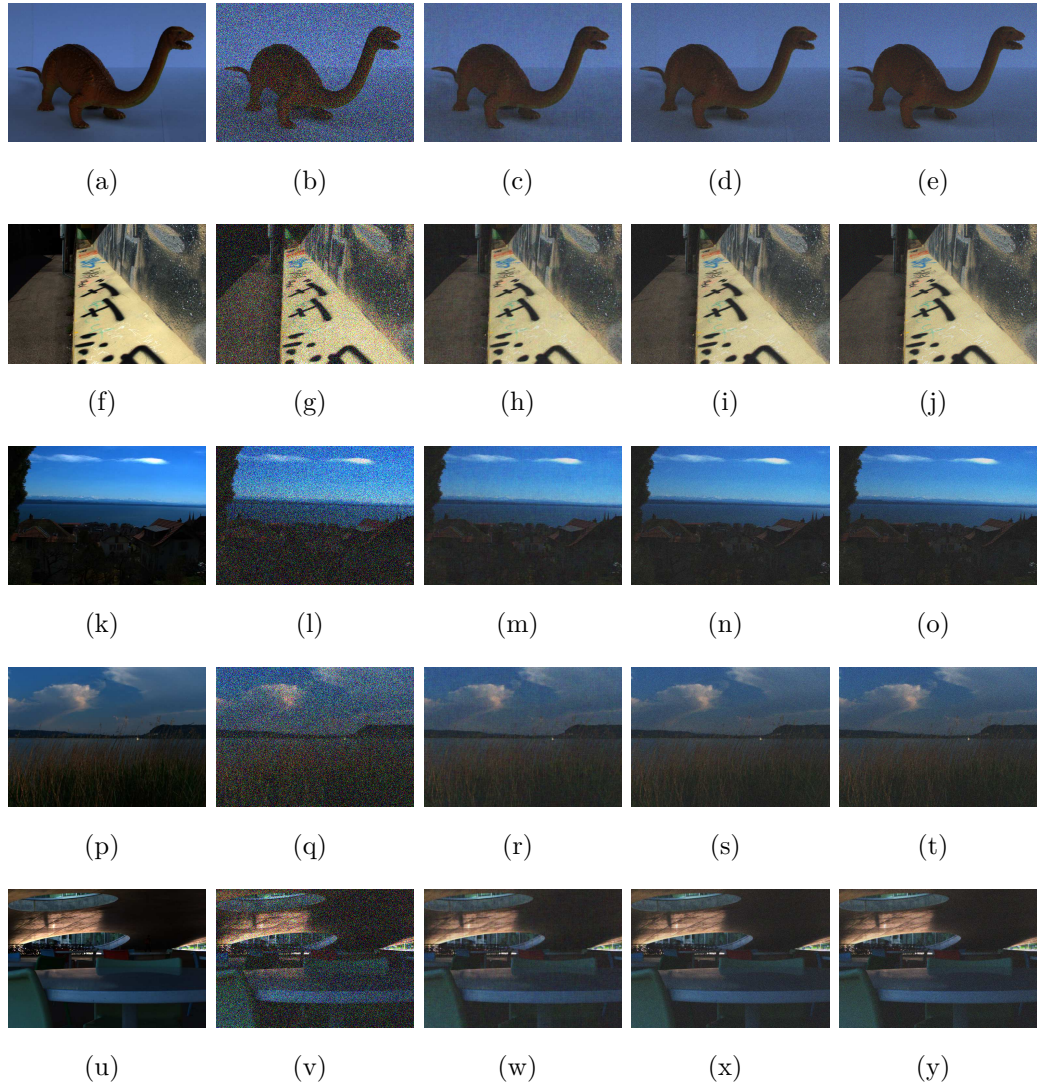


Figure D.2: SAIs of LFs from EPFL dataset for denoising with  $\sigma = 0.3$ . From left, first column: original image, second column: noisy image, third column: output of proposed method, fourth column: output of 4-D hyperfan filter, fifth column: output of 4-D planar filter, (a)-(e) Diplodocus (f)-(j) Graffiti (k)-(o) Houses & Lake (p)-(t) Reeds (u)-(y) Rolex Learning Center.

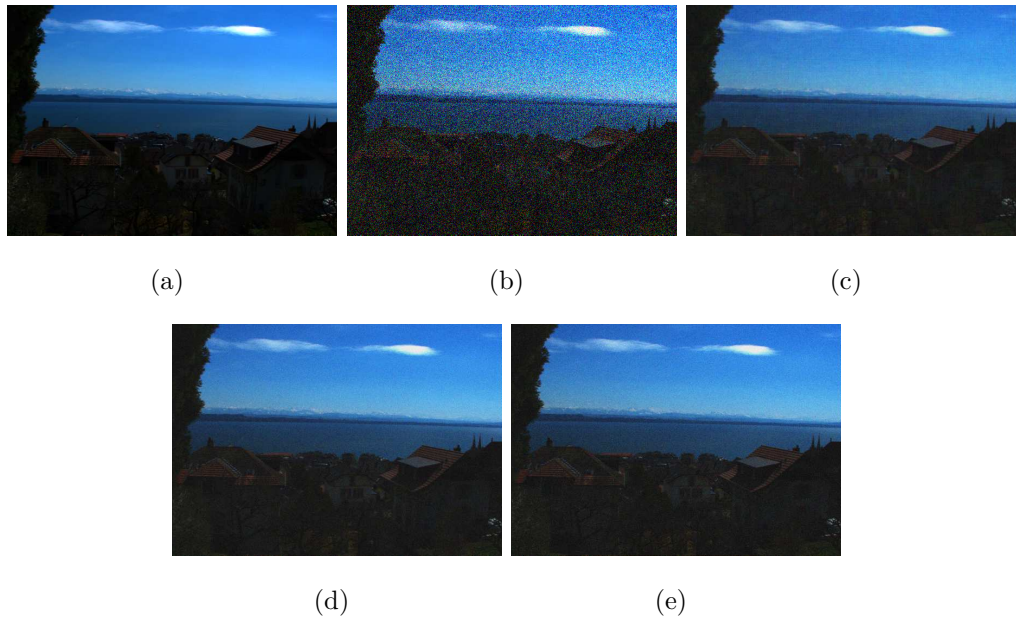


Figure D.3: SAIs of Houses & Lake LF for denoising with  $\sigma = 0.2$ . (a) original image, (b) noisy image, (c) output of proposed method, (d) output of 4-D hyperfan filter, (e) output of 4-D planar filter

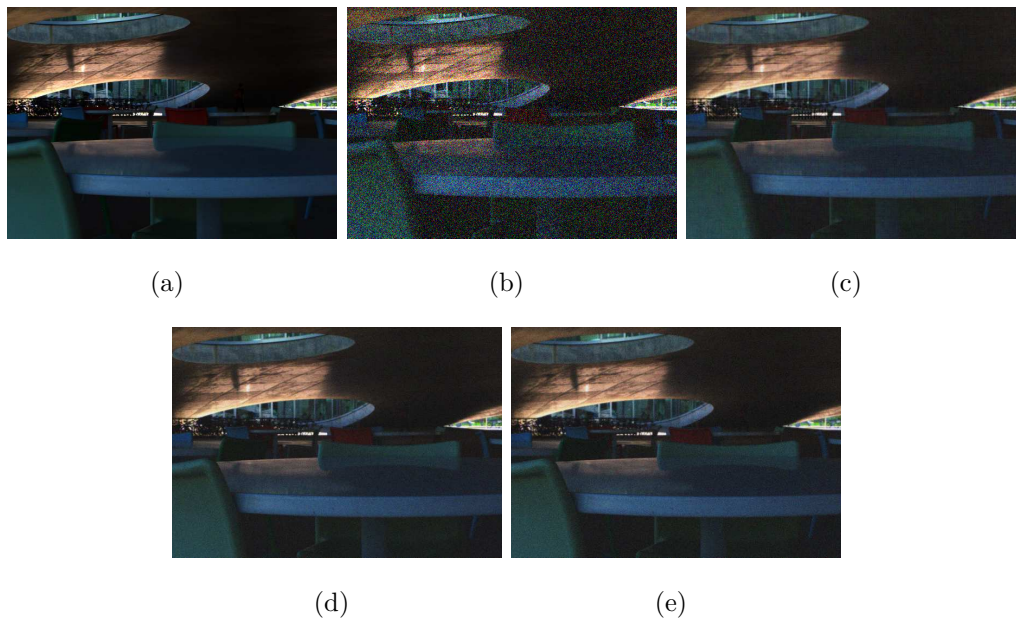


Figure D.4: SAIs of Rolex Learning Center LF for denoising with  $\sigma = 0.2$ . (a) original image, (b) noisy image, (c) output of proposed method, (d) output of 4-D hyperfan filter, (e) output of 4-D planar filter