Multi-dimensional digital signal integration with applications in image, video and light field processing

by

Ioana Speranţa Sevcenco
B.Sc., University of Bucharest, Romania, 2002
M.Sc., University of Bucharest, Romania, 2004
M.A.Sc., University of Victoria, Canada, 2012

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Multi-dimensional digital signal integration with applications in image, video and light field processing

by

Ioana Speranţa Sevcenco
B.Sc., University of Bucharest, Romania, 2002
M.Sc., University of Bucharest, Romania, 2004
M.A.Sc., University of Victoria, Canada, 2012

Supervisory Committee

Dr. Pan Agathoklis, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Wu-Sheng Lu, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Daniela Constantinescu, Outside Member
(Department of Mechanical Engineering)

ABSTRACT

Multi-dimensional digital signals have become an intertwined part of day to day life, from digital images and videos used to capture and share life experiences, to more powerful scene representations such as light field images, which open the gate to previously challenging tasks, such as post capture refocusing or eliminating visible occlusions from a scene. This dissertation delves into the world of multi-dimensional signal processing and introduces a tool of particular use for gradient based solutions of well-known signal processing problems. Specifically, a technique to reconstruct a signal from a given gradient data set is developed in the case of two dimensional (2-D), three dimensional (3-D) and four dimensional (4-D) digital signals. The reconstruction technique is multiresolution in nature, and begins by using the given gradient to generate a multi-dimensional Haar wavelet decomposition of the signals of interest, and then reconstructs the signal by Haar wavelet synthesis, performed on successive resolution levels.

The challenges in developing this technique are non-trivial and are brought about by the applications at hand. For example, in video content replacement, the gradient data from which a video sequence needs to be reconstructed is a combination of gradient values that belong to different video sequences. In most cases, such operations disrupt the conservative nature of the gradient data set. The effects of the non-conservative nature of the newly generated gradient data set are attenuated by using an iterative Poisson solver at each resolution level during the reconstruction. A second and more important challenge is brought about by the increase in signal dimensionality. In a previous approach, an intermediate extended signal with symmetric region of support is obtained, and the signal of interest is extracted from it. This approach is reasonable in 2-D, but becomes less appealing as the signal dimensionality increases. To avoid generating data that is then discarded, a new approach is proposed, in which signal extension is no longer performed. Instead, different procedures are suggested to generate a non-symmetric Haar wavelet decomposition of the signals of interest. In the case of 2-D and 3-D signals, ways to obtain this decomposition exactly from the given gradient data and the average value of the signal are proposed. In addition, ways to approximate a subset of decomposition coefficients are introduced and the visual consequences of such approximations are studied in the special case of 2-D digital images. Several ways to approximate the same subset of decomposition coefficients are developed in the special case of 4-D light field images.

Experiments run on various 2-D, 3-D and 4-D test signals are included to provide an insight on the performance of the reconstruction technique.

The value of the multi-dimensional reconstruction technique is then demonstrated by including it in a number of signal processing applications. First, an efficient algorithm is developed with the purpose of combining information from the gradient of a set of 2-D images with different regions in focus or different exposure times, with the purpose of generating an all-in-focus image or revealing details that were lost due to improper exposure setting. Moving on to 3-D signal processing applications, two video editing problems are studied and gradient based solutions are presented. In the first one, the objective is to seamlessly place content from one video sequence in another, while in the second one, to combine elements from two video sequences and generate a transparency effect. Lastly, a gradient based technique for editing 4-D scene representations (light fields) is presented, as well as a technique to combine information from two light fields with the purpose of generating a light field with more details of the imaged scene. All these applications show that the developed technique is a reliable tool for gradient domain based solutions of signal processing problems.

Table of Contents

List of Tables

List of Figures

List of Acronyms

1-D one-dimensional

2-D two-dimensional

3-D three-dimensional

4-D four-dimensional

MSE mean squared error

PSNR peak signal to noise ratio

RGB red green blue

SNR signal to noise ratio

SSIM structural similarity index measure

$\mathbf{YC_bC_r}$ luminance, blue and red chrominance

ACKNOWLEDGEMENTS

It is without a doubt that without the guidance and support of my supervisor, Dr. Pan Agathoklis, I would not be here. I am thankful to Dr. Agathoklis for all our insightful discussions and for his continued strive to make me think about what is the most important point that I am trying to achieve, and work towards it. This is an invaluable skill that often goes untaught.

My gratitude also goes towards Dr. Wu-Sheng Lu, whose endless energy and enthusiasm for research and teaching inspired me and will continue to do so for years to come. Special thank you's to the staff and IT technical support I interacted most with: Ms. Janice Closson, Mr. Dan Mai, Kevin Jones, Erik Laxdal and Matt Cormie.

Thanks also go towards those who were by my side and motivated me to complete this work: Derek, my mother and step-father, my brother and sister-in-law, and my father.

Finally, I would like to thank my supervisory committee for their insightful comments on the work reported in this thesis and acknowledge with immense gratitude the importance of the funding received from the University of Victoria, from the Natural Sciences and Engineering Research Council of Canada, and from Mr. John Montalbano. Having this funding not only helped me focus on my studies, but also inspired me to give back to the community.

DEDICATION

To all the people who believed in me.

Chapter 1

Introduction

1.1 Motivating applications

Reconstructing a signal from gradient data (first order derivatives of a signal) plays an important role in many interesting applications from adaptive optics [5], image processing [6], [7], [8] or video processing [9], [10]. This is because in some of these applications, such as adaptive optics, gradient values are available instead of signal values and in order to interpret and analyze the data, the signal has to be recovered from the gradient data. In other applications, namely those that will ultimately be evaluated by the human visual system, the gradient domain provides an interesting solution domain. Any kind of gradient manipulation, however, requires a means to recover a meaningful signal from gradient data and therefore there is a continued need for a robust integration technique, that scales well with signal dimensionality, in terms of resources, while maintaining good visual quality.

Signal reconstruction from gradient data has been reformulated as a Poisson equation in [6], and a number of techniques have been developed for signal reconstruction. A representative class of Fourier based techniques [11] with typical complexity $O(NlogN)$, where N is the number of unknowns, have been presented. An algebraic approach based on graph theory is proposed in [12]. Another class consists of iterative solvers [13], which include methods such as Jacobi and Gauss-Seidel, but these are typically slow to converge on large scale problems if a good initial point is not provided. A very popular class of fast solvers is the multigrid approach [14], which solves the Poisson problem on diagonally oriented grids, and uses an iterative Poisson solver to smooth the error at each scale and perform the interpolation to obtain solution

estimates on the finest grid. Multigrid techniques are $O(N)$.

A wavelet based reconstruction technique was proposed in [15], and used in the context of adaptive optics. This approach alos has $O(N)$ complexity, and was developed and adapted for image [16], and video [10] processing applications. This method is based on obtaining the Haar wavelet decomposition directly from the gradient data. The signal can then be obtained from this wavelet decomposition. This method deals with a non-square (in 2-D) or non-cube (in 3-D) signal by expanding the non-square signals to a square (or cube) of appropriate size. Although this expansion yields satisfactory results, it may require large amounts of additional memory, as data size increases.

1.2 Contribution of Dissertation

The main contribution of this dissertation is devising a robust integration technique that can be used to reconstruct 2-D, 3-D and 4-D signals from some gradient data set. The algorithm is similar in spirit to earlier work reported in [15], [16], and [10]. The algorithm scales well with signal dimensionality while maintaining the quality of the results and functions at its best when there are large discrepancies between signal dimensions.

The usefulness of the devised techniques will be illustrated throughout the dissertation by presenting a number of multi-dimensional signal processing applications. In Chapter 2, which deals with two dimensional signals, the algorithm is incorporated in the design of a new image fusion algorithm. In Chapter 3, which focuses on three dimensional signals, the algorithm is used in two video editing applications: one with the purpose of video content replacement and the other one for creating a transparent layering of two different videos. In Chapter 5, the algorithm is used in two light field applications: one where the objective is fusing together content from two light fields, and the second one for light field content replacement.

1.3 Outline of Dissertation

Chapter 2 reveals the details of an algorithm designed to reconstruct two dimensional signals from a gradient data set. The algorithm is then used in a gradient based image processing application and shown to yield good results. In Chapter 3, an algorithm is devised to perform the reconstruction task for three dimensional signals, and it is

then used in two video processing applications. Chapter 4 lays the groundwork for gradient based light field processing algorithms, by developing what is believed to be the first algorithm that is able to recover a 4-D signal from a given gradient. An analysis is performed on a number of light field datasets to verify the robustness of the reconstruction technique. In Chapter 5, two gradient based light field applications are developed with the help of the algorithm outlined in Chapter 4. Chapter 6 summarizes the main contributions of this dissertation and presents several directions for future research.

Chapter 2

Two dimensional signal reconstruction from gradient data

2.1 Chapter outline

An efficient wavelet-based algorithm to reconstruct signals from gradient data is proposed in this chapter. The technique is motivated by digital image processing applications that can successfully be handled in the gradient domain and is developed to efficiently address the general case of rectangular images (i.e., images with differences between sizes).

The motivation behind the developments of this chapter is given in Section 2.2. Existing reconstruction techniques are reviewed in Section 2.3. The notation used in the remainder of this chapter is introduced in Section 2.4. In Section 2.5, the details of a technique designed to reconstruct two dimensional, rectangular signals from gradient data are revealed. The experiments presented in Section 2.6 demonstrate the advantage of the newly devised reconstruction technique over existing techniques. Section 2.7 illustrates the usefulness of the newly devised tool by employing it in an image fusion application. Section 2.8 reviews the content of this chapter and provides a first round of concluding remarks and recommendations.

2.2 Motivation for 2-D study

Signal reconstruction from gradient data is an integral part of many interesting applications in adaptive optics [5] or digital image processing [6], [7], [8]. The following

paragraphs glance into some of these applications and motivate the interest for developing gradient domain based techniques.

Highly sophisticated telescopes enable celestial observations from Earth. The Earth's atmosphere, however, introduces distortions that degrade observations. To address this problem, adaptive optics advanced an ingenious solution that relies on surface reconstruction from gradient measurements. A laser beam shot from an observation point on the Earth towards the atmosphere acts as a star and generates a wavefront that travels back to Earth. Sensors then measure the slope of the incoming wavefront and reconstruct a wavefront profile that estimates the distortion introduced by the atmosphere. This information is then used to generate the opposite wavefront and model a deformable mirror, whose role is to correct the image received by the astronomical telescope.

In digital image processing, the gradient domain is often times the preferred solution domain for applications such as image editing, image stitching, or image fusion. Traditional image processing frameworks involve one or more digital images, which are stored in a computer's memory as two dimensional (2-D) arrays of numbers. Approximations of discrete versions of the image gradient are obtained and a series of mathematical operations are performed on the partial derivatives values to generate a gradient with properties driven by the application at hand. Working in the gradient domain, however, requires a means to reconstruct the image from given gradient data sets, and an efficient way to achieve this is proposed in this chapter.

2.3 Existing techniques

Signal reconstruction from gradient data has been reformulated as a Poisson equation [6], and several methods have been developed to solve the problem in this context. A well known class of solvers tackle the problem in the Fourier domain [11]. The computational complexity of these techniques is $\mathcal{O}(NlogN)$, with N being the number of unknowns, i.e. the number of signal values to be reconstructed. An algebraic solver based on graph theory is proposed in [12]. Iterative solvers [13], such as Jacobi or Gauss-Seidel were also proposed, but these typically tend to converge slowly on large scale problems. Multigrid techniques [14] were developed as improved, more efficient versions of iterative solvers. These techniques solve the problem on a hierarchy of grids, sometimes referred to as resolution levels or scales, and use iterative Poisson solvers to smooth the error at each resolution. Solution estimates on finer grids are

found by interpolation. The computational complexity of multigrid techniques is $\mathcal{O}(N)$.

The work presented in this chapter is wavelet based, and therefore multiresolution in nature with computational complexity $\mathcal{O}(N)$. It is inspired by the reconstruction from gradient technique developed in [15], [16]. The advantage of the approach presented in this dissertation over earlier developments [15], [16] is that images with non-square region of support are reconstructed in a more efficient way. An example is presented next to illustrate the main difference between the work presented in this chapter and the developments in [15], [16].

Figure 2.1a shows an example of a rectangular grayscale image[1] with size 205×400. Gradient based algorithms typically begin with obtaining the two directional components of the image gradient, depicted in Figure 2.1b. Processing is then done on the directional derivative values and at the end an image needs to be recovered from this artificially generated gradient data set. Earlier developments [15], [16], devise an integration technique which begins by finding the Haar wavelet decomposition of the desired image from the gradient data. As such, the two gradient components are constrained to be square matrices, with dimensions power of two. This constraint is addressed by expanding the given gradient components to square matrices of appropriate size (in this case, to $512 \times 512 \times 2$). The extended versions of the gradient components are shown in Figure 2.1c. An extended version of the image is then obtained, and is shown in Figure 2.1d. The desired image is extracted from the extended image at the end. Although this approach leads to satisfactory results, it requires non-negligible amounts of additional memory. The new reconstruction technique, introduced in Section 2.5, avoids increasing the size of the given gradient data, and generates a modified wavelet decomposition, which is then used to recover the image.

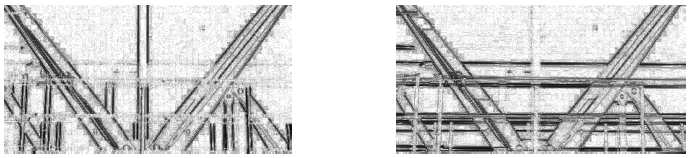
2.4 Notation

Notation and background information relevant to the tool we are developing are briefly reviewed in this section.

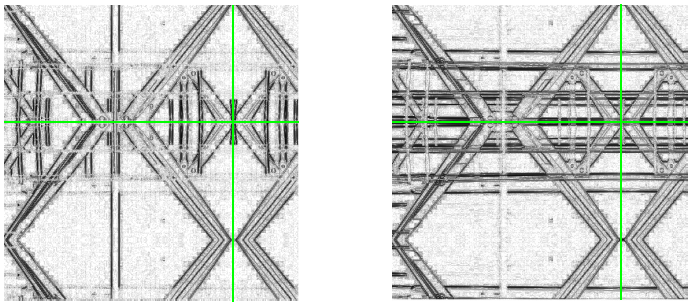
Directional downsampling (subsampling) by an integer factor n along the x or y direction is denoted $\downarrow_{n,x}$ or $\downarrow_{n,y}$, respectively. For example, downsampling a 2-D signal Φ by 2 along direction x is denoted $\downarrow_{2,x} \Phi$. Directional upsampling by an

[1] Test image obtained from MATLAB demo image **gantrycrane**

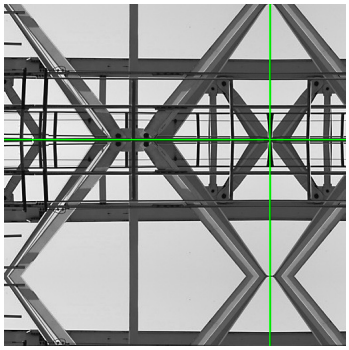
(a) Test image



(b) Gradient components



(c) Extended gradient components



(d) Extended reconstructed image

Figure 2.1: Test image, its gradient and their extended counterparts. In the gradient images, higher magnitude values are depicted in black. In the extended gradient and reconstructed image, the original data is situated in the top left, the remaining is obtained by mirroring. The green lines depict the first symmetry axis. In some cases, a second mirroring is necessary to fill the square. Here, a second mirroring operation is performed in the vertical direction

integer factor n along the x or y direction is denoted $\uparrow_{n,x}$ or $\uparrow_{n,y}$, respectively.

When the downsampling or upsampling operations are performed along both x

and y directions, the subscript indicating the direction is dropped, for a more concise notation. For example, downsampling a signal Φ by a factor of 4 along directions x and y is denoted $\downarrow_4 \Phi$.

The Haar wavelet analysis filters are relevant to the algorithm developed in this chapter and are reviewed here:

$$H_L(z) = \frac{1}{\sqrt{2}}\left(1 + z^{-1}\right) \tag{2.1}$$

$$H_H(z) = \frac{1}{\sqrt{2}}\left(1 - z^{-1}\right) \tag{2.2}$$

The discrete approximations of the partial derivatives of a 2-D image Φ can be obtained using the Haar wavelet highpass analysis filter $H_H(z)$:

$$\Phi_x = \sqrt{2}\Phi H_H(z_x) \tag{2.3}$$

$$\Phi_y = \sqrt{2}\Phi H_H(z_y) \tag{2.4}$$

where, $\Phi H(z)$ denotes one dimensional filtering of the two-dimensional signal Φ with the one dimensional filter given by its transfer function $H_H(z)$. The subscripts x and y in the expression $\Phi H(z_x)$, $\Phi H(z_y)$ indicate that the filtering is performed along direction x or y, and the two directions are assumed orthogonal.

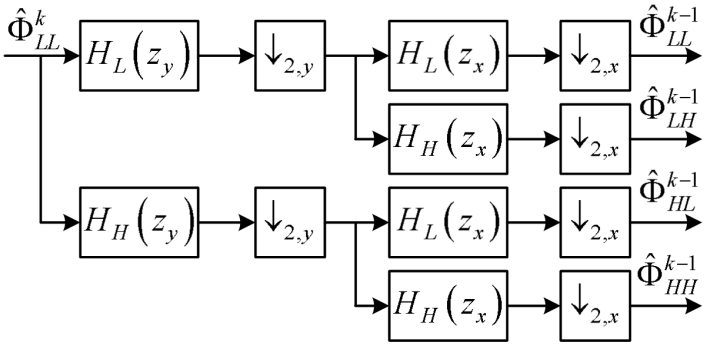


Figure 2.2: One step in the (Haar) wavelet decomposition of a 2-D signal. The convention is that $\hat{\Phi}^k_{LL}$ denotes the full resolution signal Φ when $k = M$ and all other $\hat{\Phi}^k_{LL}$, for which $0 \leq k \leq M-1$ are approximation wavelet coefficients at lower resolutions.

2.5 Two dimensional signal reconstruction from gradient

Let Φ be an unknown 2-D signal, with size $2^M \cdot 2^N$ with $M \leqslant N$ two strictly positive integers. The signal value at a point (y, x) is denoted $\Phi(y, x)$ with the convention that y and x are orthogonal directions. The first order discrete directional derivatives of the unknown signal Φ are denoted by Φ_y and Φ_x, and let m be the mean value of the unknown signal Φ. An algorithm is described here to obtain the 2-D signal Φ from the given gradient data Φ_y and Φ_x.

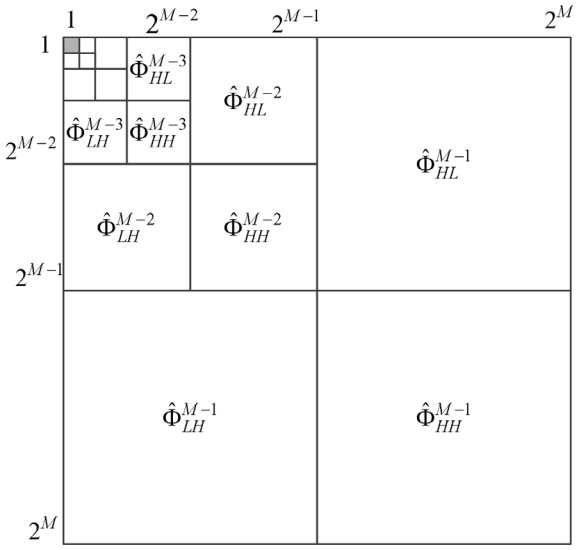
2.5.1 Analysis step: detail subbands of the wavelet decomposition

The first step of the reconstruction algortihm is to obtain the Haar wavelet decomposition of the signal directly from the derivatives. In Figure 2.3, the Haar wavelet decompositions of a square and non-square 2-D signal are illustrated, side by side. For a non-square signal with size $2^M \times 2^N$, the equations for finding the "low-high", "high-low" and "high-high" subbands at all resolutions of the Haar wavelet decomposition from the given gradient components are the same as those for a square signal. For more details, the reader is referred to [16], with the mention that, as the signal is not a square matrix, the maximum number of levels in the decomposition is M (if $M < N$).

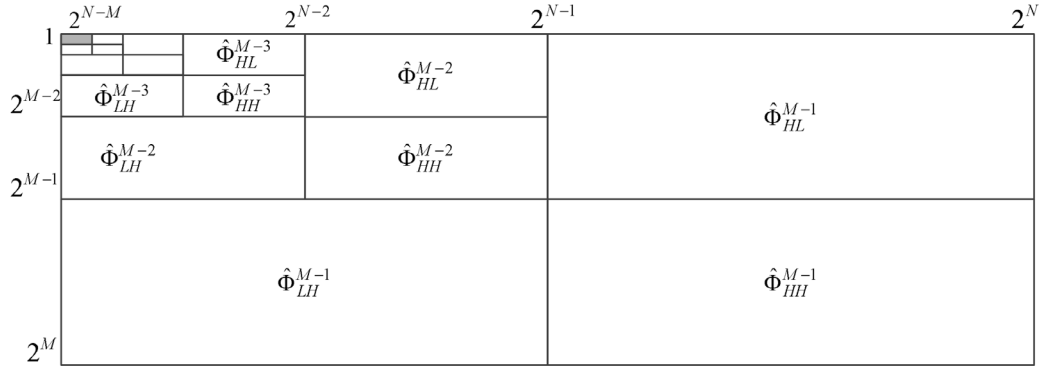
What is different in the non-square case is the way in which the "low-low" subband at the coarsest resolution of the Haar wavelet decomposition is found. These subband coefficients are referred to as the coarsest resolution subband. The following section details how these coefficients can be found.

2.5.2 Analysis step: approximation subband at lowest resolution

The coarsest resolution approximation subband of the Haar wavelet decomposition of Φ is denoted $\hat{\Phi}^0_{LL}$ and is a signal with size $1 \times 2^{N-M}$. It is the output obtained by repeating the process shown on the top branch of Figure 2.2 and is the highlighted entry in the decomposition shown in Figure 2.3b. If one is given the signal, this subband is obtained by successive filtering of the signal with the analysis lowpass



(a) Haar wavelet decomposition of 2-D signal (square case)



(b) Haar wavelet decomposition of 2-D signal (rectangular case)

Figure 2.3: Haar wavelet decomposition of a 2-D signal - square versus rectangular case

filter $H_L(z)$ followed by downsampling by 2. This subband is thus proportional to the partial sums of all elements in consecutive regions of Φ, as illustrated in Fig. 2.4. The objective of this work is to find this subband not from the signal, but rather from the signal derivatives and the mean value of the signal.

In the square case, the coarsest resolution subband is a scalar proportional to the mean value of the signal. In particular, the coarsest resolution subband is equal to $2^M \cdot m$, where m is the mean value of the signal. This can be inferred from studying Figure 2.2, and taking into account that a complete wavelet decomposition of a 2-D square signal with size $2^M \times 2^M$ is obtained in M steps, each one consisting of a scaled averaging operation followed by subsampling by two.

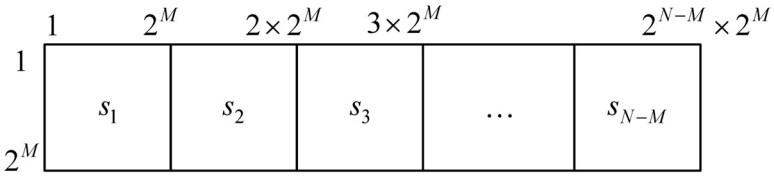


Figure 2.4: Illustration of the connection between the rectangular signal Φ with size $2^M \times 2^N$, with $M \leq N$, and the coarsest resolution subband $\hat{\Phi}^0_{LL} = [s_1, s_2, ..., s_{N-M}]$

In the general case of non-square signals, of interest here, the coarsest resolution approximation subband is no longer a scalar, but rather an array of numbers, shown highlighted in the top left part of Figure 2.3b. Finding $\hat{\Phi}^0_{LL}$ amounts to finding the $N - M$ partial sums of signal regions illustrated in Figure 2.4. The procedure for finding $\hat{\Phi}^0_{LL}$ in the general case is described below:

Step 1. Compute:

$$c_x = A_x \Phi_x^T B_x \tag{2.5}$$

where:

$$A_x = \begin{pmatrix} 1 & 2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 2 & 1 \end{pmatrix} \in \mathbb{R}^{\left(2^{N-1}-1\right)\times\left(2^N-1\right)}$$

and

$$B_x = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \in \mathbb{R}^{2^M \times 1}$$

Step 2. Compute:

$$u = \frac{1}{2^M} A^{-1} v \tag{2.6}$$

where:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ -1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{2^{N-1} \times 2^{N-1}}$$

and

$$v = \begin{bmatrix} m \cdot 2^{M+N} \\ c_x \end{bmatrix} \in \mathbb{R}^{2^{N-1} \times 1}$$

Step 3. Compute:

$$\hat{\Phi}^0_{LL} = u^T B \tag{2.7}$$

where:

$$B = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_k \end{bmatrix}^T$$

$$k = 2^{N-M}$$

and

$$\mathbf{B}_p = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{2^{M-1} \times 2^{N-M}}$$

in which the p^{th} column is an all-ones vector with 2^{M-1} elements.

A small size example that show how Steps 1–3 above can be used to generate the coarsest resolution approximation subband is included in Appendix A of this dissertation.

The benefit of the algorithm summarized above is that it produces the $\hat{\Phi}^0_{LL}$ entry exactly, from the signal derivatives and its mean value, and thus the complete wavelet decomposition of the signal without the need to extend the signal size to the same nearest power of two along both dimensions. Step 2 in the algorithm, however, involves a matrix inversion, and it is well known that such an operation influences algorithm performance, depending on the size of the matrix to be inverted. If solution

accuracy is desired, then the matrix should be inverted, otherwise, various algorithms can be employed to obtain an approximate solution, depending on the properties of matrix A.

2.5.3 Synthesis step

In the synthesis step, the signal is obtained from the Haar wavelet decomposition by wavelet synthesis, with the possibility of including a basic iterative Poisson solver at each resolution, as described in [16]. The Poisson solver used is also included in the Appendix B of this Dissertation. Including the iterative Poisson solver is important and recommended in applications such as image or video editing in the gradient domain, where the gradient data has been altered and the zero-curl condition is not satisfied.

2.6 Performance evaluation

This section analyzes the performance of different reconstruction algorithms and the role of the coarsest resolution subband coefficients on the quality of signals reconstructed from gradient data. The considered signals are non-square. The comparison will be between algorithms are listed in Table 2.1 and described below:

Algorithm 1: the method proposed in Section 2.5

Algorithm 2: the method proposed in [16] where non-square gradient signals were extended to the nearest square

Algorithm 3: the method proposed in Section 2.5, but assuming all elements of the coarsest resolution subband are equal to zero.

Algorithm 3 is particularly interesting in practical applications such as image or video editing, as it avoids both the use of additional memory required for the extension

	Alg. 1	**Algorithm 2**	**Algorithm 3**
Source	Sec. 2.5.1, 2.5.2, 2.5.3	Ref. [16]	Sec. 2.5.1, 2.5.3
Extension performed	No	Yes	No
Average value used	Yes	Yes	Yes
Poisson solver used	Yes	Yes	Yes

Table 2.1: Analyzed algorithms summary

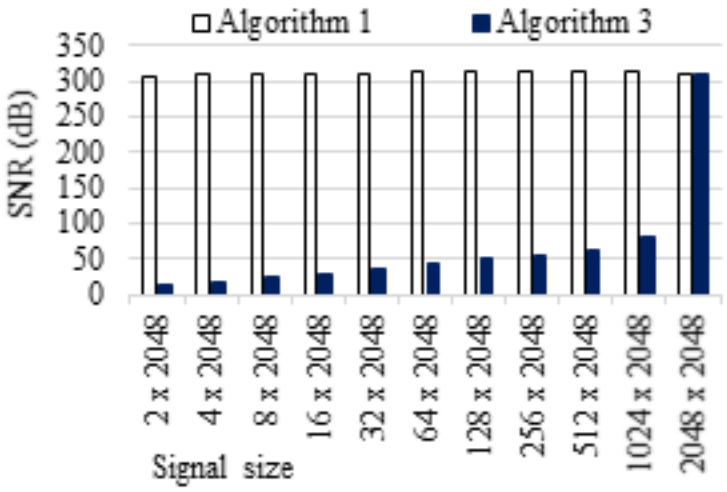


Figure 2.5: Performance evaluation in terms of solution accuracy

to a square and the computations to obtain the correct coarsest resolution subband (see the discussion regarding the matrix inversion in Section 2.5). Results show that although this approach leads to low accuracy reconstructions, the signals have a good visual quality provided that the iterative Poisson solver is included in the synthesis step.

Determining the quality of the devised technique is done by monitoring two criteria: solution accuracy and efficiency. The signal to noise ratio (SNR), measured in dB, between the original signal and the estimate obtained from the derivatives and the mean value of the signal, via Algorithms 1 – 3, is computed to rank solution accuracy, while the CPU time required for a MATLAB implementation to produce a solution, is recorded as a measure of algorithm efficiency.

Reconstruction accuracy was first considered for Algorithms 1 and 3, and the results are shown in Figure 2.5. The reconstruction accuracy of Algorithm 2 is similar to that of Algorithm 1. In Algorithm 3, all entries of $\hat{\Phi}^0_{LL}$ are set to zero before wavelet synthesis, and the average value of the reconstruction is corrected at the end, to match the given average. This corresponds to the partial sums illustrated in Figure 2.4 all having the same value. As can be seen from Figure 2.5, obtaining the values of $\hat{\Phi}^0_{LL}$ exactly from the input data results in a very high SNR in the reconstructed signal, and, as expected, leaving the coarsest resolution "low-low" subband coefficients zero before synthesis and adjusting the signal average value at the end significantly lowers the SNR of the reconstructed signal.

Run times of MATLAB implementations of the three reconstruction algorithms

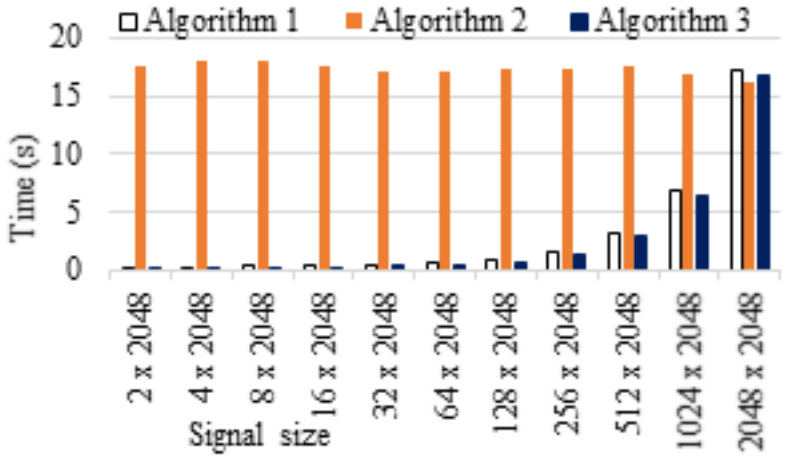


Figure 2.6: Performance evaluation in terms of efficiency. Algorithm 1: reconstruction without signal extension, by finding $\hat{\Phi}^0_{LL}$ as described in Section 2.5.2; Algorithm 2: signal reconstruction with gradient extension to nearest square, as in [16]; Algorithm 3: signal reconstruction without gradient extension, by setting the elements of $\hat{\Phi}^0_{LL}$ to zero before synthesis

were compared. As Figure 2.6 reveals, the performance is similar for square signals, as signal extension is not performed in this case. The advantage of the new approach (i.e. Algorithms 1 and 3) becomes clearer when dealing with non-square signals. In this case, a solution is produced significantly faster by either Algorithm 1 or 3 than by Algorithm 2, where signal extension is performed. These experiments offer an insight on the main trade-off that characterizes the new algorithm, and a recommendation can be made. If signal accuracy is of primary importance for the application at hand, the coarsest resolution subband coefficients should be computed exactly before wavelet synthesis. If memory requirements and speed are more important, then the coarsest resolution subband coefficients can be approximated, and the result is still visually acceptable if the Poisson solver is included in the wavelet synthesis step.

2.6.1 Visualizing the effect of LL at lowest resolution in the non square case

Let us now take a closer look at the reconstruction error in Algorithm 3. The example in Figure 2.7 helps visualize this error. An image with size 64×256 was obtained from image "threads" and is shown in Figure 2.7a. The image reconstructed from its gradient and mean value using Algorithm 1 is shown in Figure 2.7b. As expected, this is an accurate reconstruction of the original image. The image reconstructed using

Algorithm 2, i.e. , without the exact calculation of the coarsest resolution subband coefficients, is shown in Figure 2.7c. Clearly, there are noticeable visual artifacts caused by errors in the reconstruction. These artifacts are a direct consequence of not obtaining the exact coarsest resolution "low-low" subband coefficients of the Haar wavelet decomposition of the image, from the gradients. The reconstruction in Figure 2.7d is obtained without the exact coarsest resolution "low-low" subband coefficients but with the incorporation of an iterative Poisson solver at each resolution during wavelet synthesis. Including the Poisson solver improves the visual quality of the reconstruction by removing the vertical lines in Figure 2.7c. The objective quality of each approximation was also evaluated by comparison with the original image, in terms of SNR and SSIM [17]. The reported SNR values agree with the noticeable increase in visual quality (from Figure 2.7c to 2.7d). It is interesting to note that SSIM seems to indicate that the result in Figure 2.7d is of lower quality of Figure 2.7c, although a visual examination of the two indicates otherwise.

2.7 Application: image fusion in the gradient domain

In this section, the devised algorithm will be employed in an image processing application, to demonstrate its purpose in reconstructing images from a modified gradient data set, and therefore prove its potential in image processing and computer vision applications. A multi-exposure and multi-focus image fusion algorithm[2] is proposed. The algorithm is developed for color images and is based on blending the gradients of the luminance components of the input images using the maximum gradient magnitude at each pixel location and then obtaining the fused luminance using the reconstruction technique introduced in Section 2.5. The chrominance information of the fused image is a weighted sum of the chrominance channels of the input images. The special case of grayscale images is treated as luminance fusion. Experimental results and comparison with other fusion techniques indicate that the proposed algorithm is fast and produces similar or better results than existing techniques for both multi-exposure and multi-focus images.

[2]Part of this section was published in [18]

(a) Original image

(b) Reconstruction via Algo. 1; SNR: 300.06dB; SSIM: 1

(c) Reconstruction via Algo. 3, no Poisson; SNR: 16.47dB; SSIM: 0.91

(d) Reconstruction via Algo. 3, with Poisson; SNR: 18.09dB; SSIM: 0.85

Figure 2.7: Visualizing the role of the coarsest resolution low subband coefficients and of the Poisson solver. Top to bottom: original image; image reconstructed with exact calculation of $\hat{\Phi}^0_{LL}$ from the given data; image reconstructed by setting all elements of $\hat{\Phi}^0_{LL}$ to a constant; image reconstructed by setting all elements of $\hat{\Phi}^0_{LL}$ to a constant, followed by modified wavelet synthesis, with iterative Poisson solver included at each resolution

2.7.1 The image fusion problem

Image fusion is a technique that makes use of existing information from a stack of images to produce a single image with more visible details than any of the individual images in the stack. The need for such detailed images is present in a variety of fields, such as computer vision, medical imaging, photography and remote sensing, where either one or more imaging devices are used to generate digital scene representations.

Image fusion can be applied to multi-focus or multi-exposure images. In the multi focus case, the input images are those in which only some portion of the image is well focused, whereas other portions appear blurred. Haghighat et al. [19] proposed a multi focus image fusion technique that operates in the discrete cosine transform (DCT) domain. They compute the variance of the 8×8 DCT coefficients of each image, and the fused blocks are those having the highest variance of DCT coefficients. Song et al. [20] proposed a wavelet decomposition-based algorithm for multi-focus image fusion. They fuse the wavelet coefficients using an activity measure which depends on the gradients of the wavelet coefficients. A multiresolution approach was also adopted in the algorithms developed by Li and Wang in [21] and by Biswas et al. in [22]. A survey on multi-focus image fusion techniques can be found in [23]. More recent research [24], [25] makes use of edge detection techniques for color image fusion.

In the multi-exposure case, the input images have different exposures. These images have details only in a part of the image while the rest of the image is either under- or over-exposed. Fusion of such images is done to integrate the details from all images into a single, more comprehensive result. Mertens et al. [26] proposed such an algorithm, in which the images are decomposed into Laplacian pyramids and then they are combined at each level using weights depending on the contrast, saturation and well-exposedness of the given images. A technique for image contrast enhancement using image fusion has been presented in [27] and is similar to [26]. In [27], the input images to the fusion algorithm are obtained from the original image after applying local and/or global enhancements. Shen et al. [28] use a probabilistic model based on local contrast and color consistency to combine multi-exposure images. Li et al. [29] fuse the multi-exposure images using a weighted sum methodology based on local contrast, brightness and color dissimilarity. They use a pixel-based method instead of a multi-resolution approach to increase the speed of execution. Kong et al. [30] propose an algorithm where the input images are first divided into blocks and then the blocks corresponding to maximum entropy are used to obtain the fused image.

The genetic algorithm (GA) is used to optimize block size, and this may require a considerable amount of time to converge.

Image fusion in the gradient domain has been recently studied by some researchers. Socolinsky and Wolff [31] proposed an image fusion approach which integrates information from a multi-spectral image dataset to produce a one band visualization of the image. They generalize image contrast, which is closely related to image gradients, by defining it for multi-spectral images in terms of differential geometry. They use this contrast information to reconstruct the optimal gradient vector field, to produce the fused image. Later, Wang et al. [32] fused the images in gradient domain using weights dependent on local variations in intensity of the input images. At each pixel position, they construct an importance-weighted contrast matrix. The square root of the largest eigenvalue of this matrix yields the fused gradient magnitude, and the corresponding eigenvector gives the direction of the fused gradient. Recently, Hara et al. [33] used an inter image weighting scheme to optimize the weighted sum of the gradient magnitude and then reconstruct the fused gradients to produce the fused image. The optimization step tends to slow down this technique. Additionally, their technique comprises a manually thresholded intra image weight saliency map, requiring user intervention. An interesting block-based approach was recently proposed by Ma and Wang in [34]. This approach is unique in the way in which it processes color images. Specifically, the RGB color channels of an image are processed together, and instead the images are split into three "conceptually independent components: signal strength, signal structure and mean intensity" [34]. This idea was inspired by the increasingly popular structural similarity index measure (SSIM) [17] developed by the same main author as an objective measure of similarity between two images.

2.7.2 The image fusion algorithm

A gradient-based image fusion algorithm is proposed here, for the fusion of both color and grayscale images. In the case of color images, one of the key ideas of the fusion algorithm proposed here is that it treats the luminance and chrominance channels of the images to be fused in a different manner. This different treatment of the channels is motivated by the fact that the luminance channel contains a major part of information about image details and contrast, whereas the chrominance channels contain only color information, to which the human visual system is less sensitive. The fusion of the luminance channels is done in the gradient domain, by taking the

gradients with the maximal magnitude of the input images at each pixel location. The luminance channel of the fused image is then obtained by integrating the fused gradients. This is done by using the technique developed in Section 2.5. An earlier version of this algorithm is known [16] to produce good results, free from artifacts, when the gradient field is a nonconservative field, as is the case when gradients from different images are combined.The chrominance components of the fused image fusion is done by taking a weighted sum of the input chrominance channels, with the weights depending on the channel intensities, which conveys information about color. Grayscale images may be dealt with in the same way as the luminance component of color images. The proposed algorithm can be applied for multi-exposure as well as multi-focus images.

2.7.2.1 Luminance fusion

As mentioned earlier, luminance fusion can be carried out on grayscale images, or on color images that are in the YCbCr color coordinate system. If the input images are in RGB representation, conversion to YCbCr should be performed first.

Luminance fusion is performed in the gradient domain. This domain choice is motivated by the fact that the image gradient depicts information on detail content, to which the human visual system is more sensitive under certain illumination conditions. For example, a blurred, over- or under-exposed region in an image will have a much lower gradient magnitude of the luminance channel than the same region in an image with better focus or exposure. This observation implies that taking the gradients with the maximal magnitude at each pixel position will lead to an image which has much more detail than any other image in the stack.

Let the luminance channels of a stack of N input images be $I = \{I_1, I_2, \ldots, I_n\}$. The image gradients according to a commonly employed discretization model for the luminance channel of the n^{th} image in the stack may be defined as:

$$\Phi_n^y(x, y) = I_n(x, y+1) - I_n(x, y) \tag{2.8}$$

$$\Phi_n^x(x, y) = I_n(x+1, y) - I_n(x, y) \tag{2.9}$$

where Φ_n^y and Φ_n^x are the gradient components in the x- and y-directions. The magnitude of the gradient may be defined as:

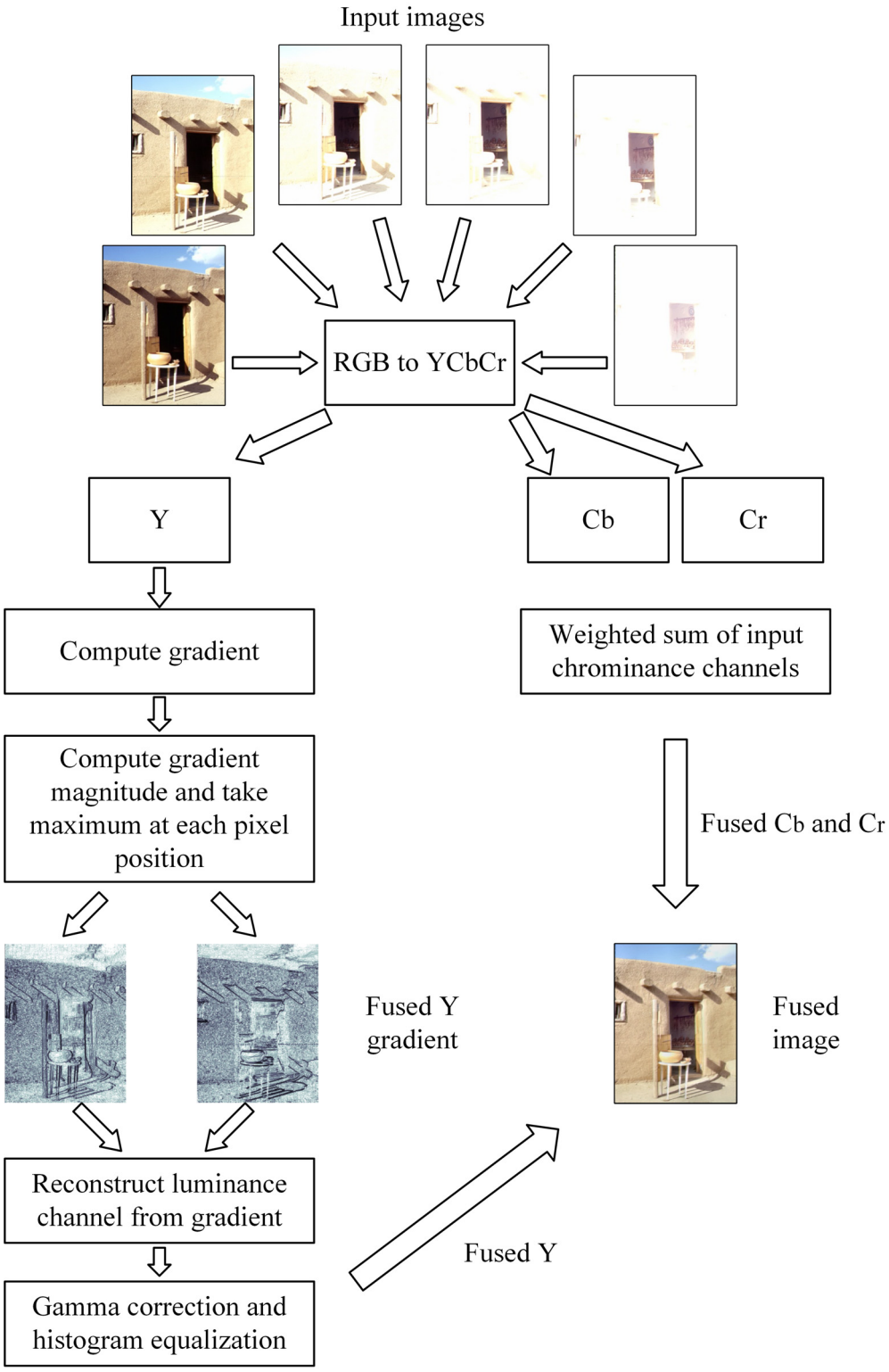


Figure 2.8: Pictorial representation of the image fusion algorithm

$$H_n(x,y) = \sqrt{\Phi_n^x(x,y)^2 + \Phi_n^y(x,y)^2} \tag{2.10}$$

Let the image number having the maximum gradient magnitude at the pixel location (x, y) be $p(x, y)$. It may be mathematically represented as:

$$\Phi^x(x,y) = \Phi^x_{p(x,y)}(x,y) \tag{2.11}$$

$$\Phi^y(x,y) = \Phi^y_{p(x,y)}(x,y) \tag{2.12}$$

So, the fused luminance gradient is $\Phi = [\Phi^x, \Phi^y]$. It may be noted that the fused luminance gradient has details from all the luminance channels from the stack and in order to get the fused luminance channel, reconstruction is required from the gradient domain. The relationship between the fused gradient (Φ) and the fused luminance channel (I) may be represented as:

$$\nabla I = \Phi \tag{2.13}$$

The image is reconstructed from the gradient domain by using the technique described in Section 2.5, with three iterations of Poisson solver at each resolution during the synthesis step. After obtaining the image from the gradient domain, some pixels may have intensity values outside the standard range of the luminance component (16–235). This is due to the fact that the fused gradient is obtained by merging multiple image gradients, and as a result, high differences between neighboring gradient values exist, leading to a reconstructed image with a high dynamic range of pixel intensities. A linear mapping of the pixel intensities of the reconstructed luminance channel can be done such that the resultant intensities lie within the required range. The caveat of this approach, however, is that it leads to a loss of contrast. For this reason, a non-linear mapping similar to gamma correction is used. The resultant image may be obtained using:

$$I(i,j) = \left(\frac{I(i,j) - \min\limits_{i,j} I(i,j)}{\max\limits_{i,j} I(i,j) - \min\limits_{i,j} I(i,j)} \right)^{\gamma} \times R_C + L \tag{2.14}$$

where $\gamma = \log_e(R_C)/\log_e(R_I)$, R_I is the range of values present in the reconstructed luminance component, $R_C = H - L$, and H and L are the maximum and minimum intensity values in the channel. In the case of the luminance component of a color

image, $H = 235$ and $L = 19$, thus $R_C = 216$. Using Eq. 2.14 generates a result with more details than the input images. At the end, local histogram equalization [35] is applied on the luminance component. This is done in order to distribute the intensities properly throughout the entire range of display. It may be noted that grayscale images can be fused in the same way as luminance component of a color image. In case of grayscale images, $H = 255$, $L = 0$, thus $R_C = 255$.

2.7.2.2 Chrominance fusion

The chrominance channel fusion is done by taking a weighted mean of the input chrominance channels. The values in the chrominance channels have a range from 16–240 and carry information about color. These channels are such that when both C_b and C_r are equal to 128, the image is visually similar to a grayscale image, and thus carries the least amount of color information. This motivates selecting the weights for the chrominance channels such that at each pixel position they depend on how far from 128 the chrominance value is. Let us denote the chrominance channels of the stack of input images by $C'_b = \{C_b^1, C_b^2, \ldots, C_b^N\}$ and $C'_r = \{C_r^1, C_r^2, \ldots, C_r^N\}$. The fused chrominance channels may be represented as follows:

$$C_b(i,j) = \sum_{n=1}^{N} w_b^n(i,j) \,.\, (C_b^n(i,j) - 128) + 128 \tag{2.15}$$

where

$$w_b^n(i,j) = \frac{|C_b^n(i,j) - 128|}{\sum_{k=1}^{N} \left|C_b^k(i,j) - 128\right|} \tag{2.16}$$

$$C_r(i,j) = \sum_{n=1}^{N} w_r^n(i,j) \,.\, (C_r^n(i,j) - 128) + 128 \tag{2.17}$$

where

$$w_r^n(i,j) = \frac{|C_r^n(i,j) - 128|}{\sum_{k=1}^{N} |C_r^k(i,j) - 128|} \tag{2.18}$$

where $|\cdot|$ returns the absolute value. If all chrominance values at a pixel position in all images from the stack are equal to 128, the corresponding weights will be zero. It may be noted that the fusion of the chrominance channels done by equations 2.15–2.18 is a pixel based approach, and is thus less computationally intensive than luminance

fusion, which is gradient based. Figure 2.8 presents a pictorial representation of the proposed algorithm.

The experiments presented in the following section indicate that the proposed algorithm works well to fuse both multi-exposure and multi-focus images.

2.7.3 Image fusion results

In this section, the performance of the proposed image fusion algorithm is evaluated on different sets of images. The results obtained using the proposed technique are compared with the results produced by four other image fusion algorithms, namely DCT [19], SVD [36] multi-exposure fusion (MEF) [26], and the gradient weighting (GrW) method [33]. The input images used in the comparison are grouped into four different classes according to the type of fusion performed (i.e., multi focus and multi-exposure, grayscale and color) and are presented in the following subsections.

The performance analysis begins with a visual comparison of the results produced by each of the studied algorithms. In passing we note that, to the best of our knowledge, this kind of evaluation (i.e., subjective evaluation) continues to dominate the chart of quality assessment measures for image fusion algorithms. The use of objective measures will be discussed later. The codes for the algorithm proposed in this paper are available at [37].

2.7.3.1 Results - multi exposure grayscale case

Two multi-exposure grayscale images named **igloo** [38] and **monument** [39] are presented in Figures 2.9a–2.9f and Figures 2.10a–2.10c, respectively. The fused results of the proposed algorithm are presented in Figures 2.9h and 2.10e, respectively. GrW [33] is an algorithm for image fusion where the authors have used multi-exposure grayscale images to test their method. It is a gradient domain fusion method and requires reconstruction to get the fused image. As the authors of the GrW algorithm have not mentioned any specific method for reconstruction, the wavelet based reconstruction procedure in Section 2.5 has been used to yield the fused image. The saliency map used by the authors of GrW is not used here, because no automated way of selecting the threshold for the map has been mentioned in their paper. The fused results produced by the GrW method are presented in Figures 2.9g and 2.10d. It may be observed from Figure 2.9 that the details inside the igloo building are more visible in the result produced using the method proposed in this paper than in the

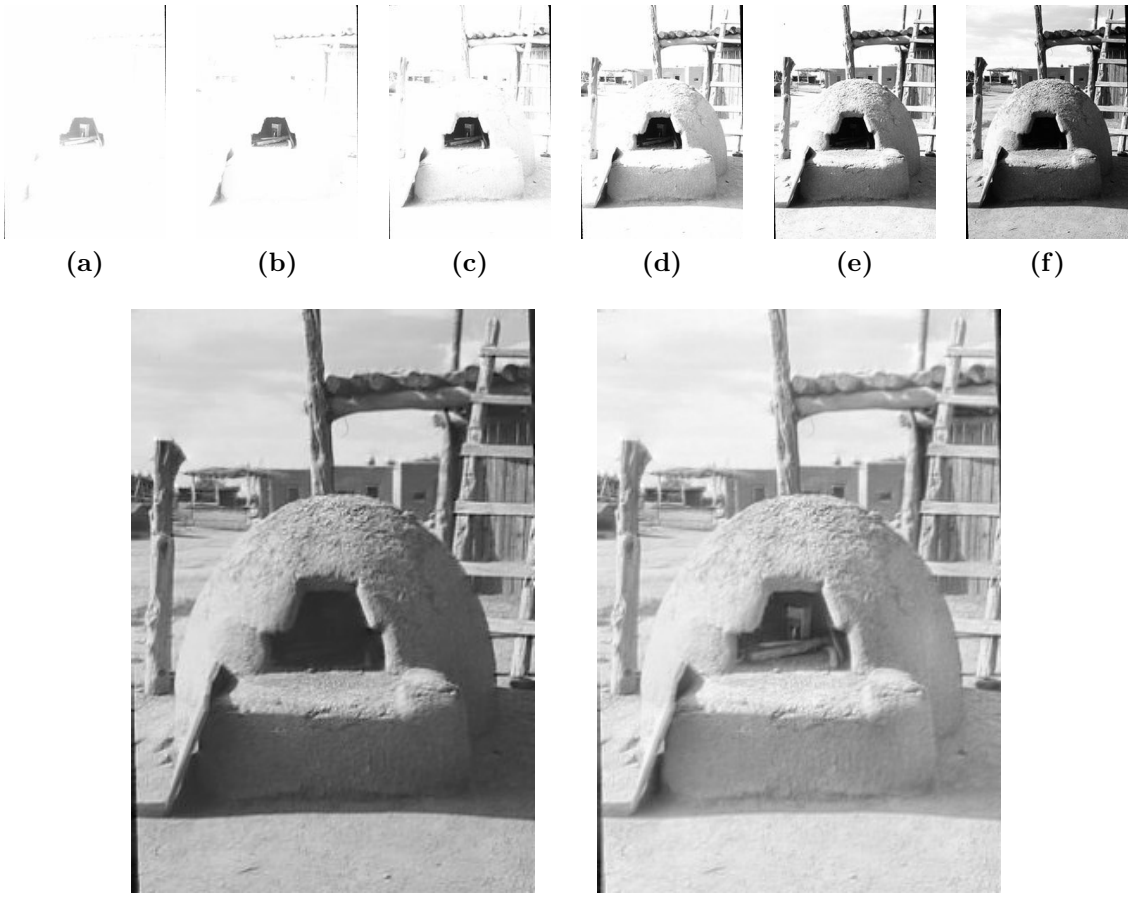


Figure 2.9: The first row contains the input images (**igloo**). The second row contains the fused image by GrW and proposed algorithm (left to right), respectively.

one produced by the GrW method. Again, in Figure 2.10, the sky-cloud portion is more visible in the image fused by the proposed algorithm than in the image fused by the GrW method.

2.7.3.2 Results - multi exposure color case

Images **Door** [40] and **house** [40] are two multi-exposure color images presented in Figures 2.11a–2.11f and Figures 2.12a–2.12d, respectively. It may be observed that for the door image, details within the door are not visible in the first input image and the details outside the door are not visible in the last input image. The proposed algorithm fuses all input images properly, as may be observed from the results presented in Figures 2.11h and 2.12f for the door and house images, respectively. A technique for multi-exposure fusion of color images presented in the literature is MEF [26]. This

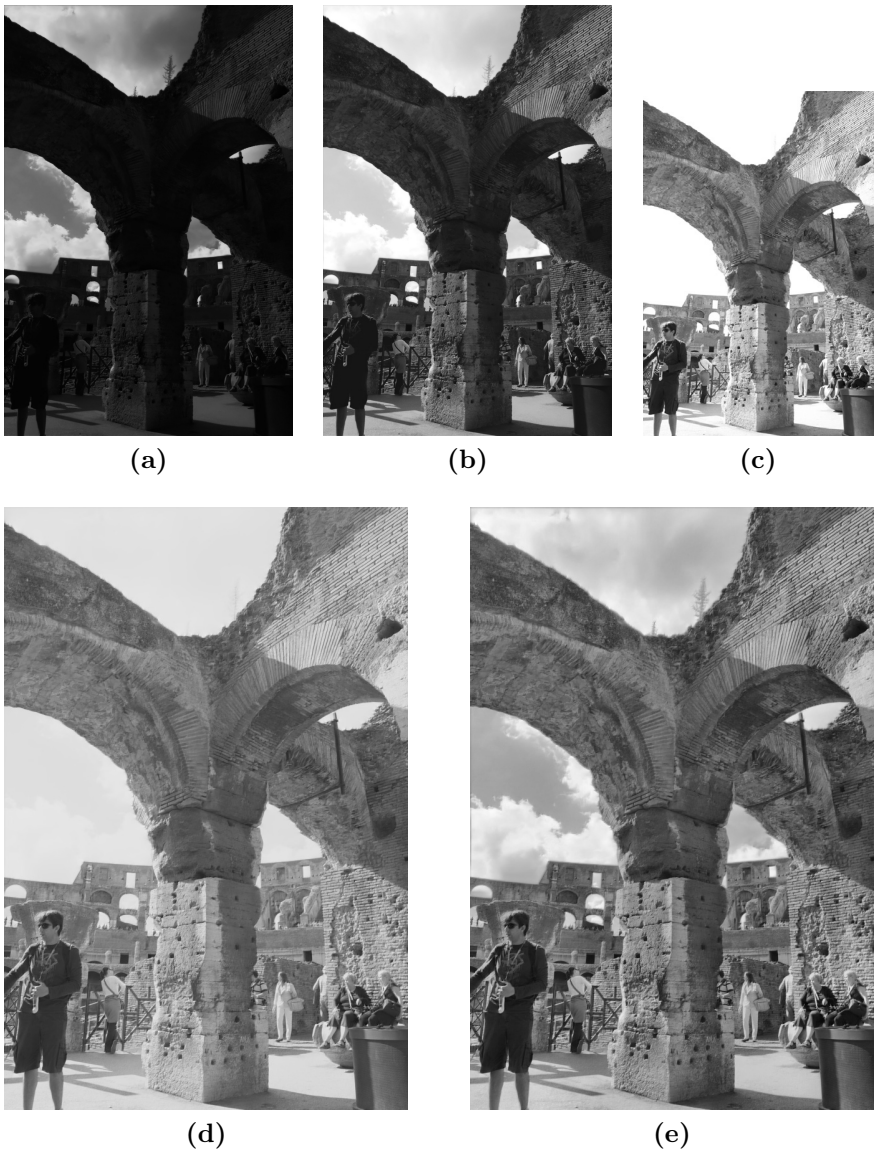


Figure 2.10: The first row contains the input images (**monument**). The second row contains the fused image by GrW and proposed algorithm (left to right).

method uses a saturation measure defined only for color images. The results produced by the MEF method are presented in Figures 2.11g and 2.12e, respectively. It can be seen that the MEF algorithm performs in a similar fashion to the proposed method.

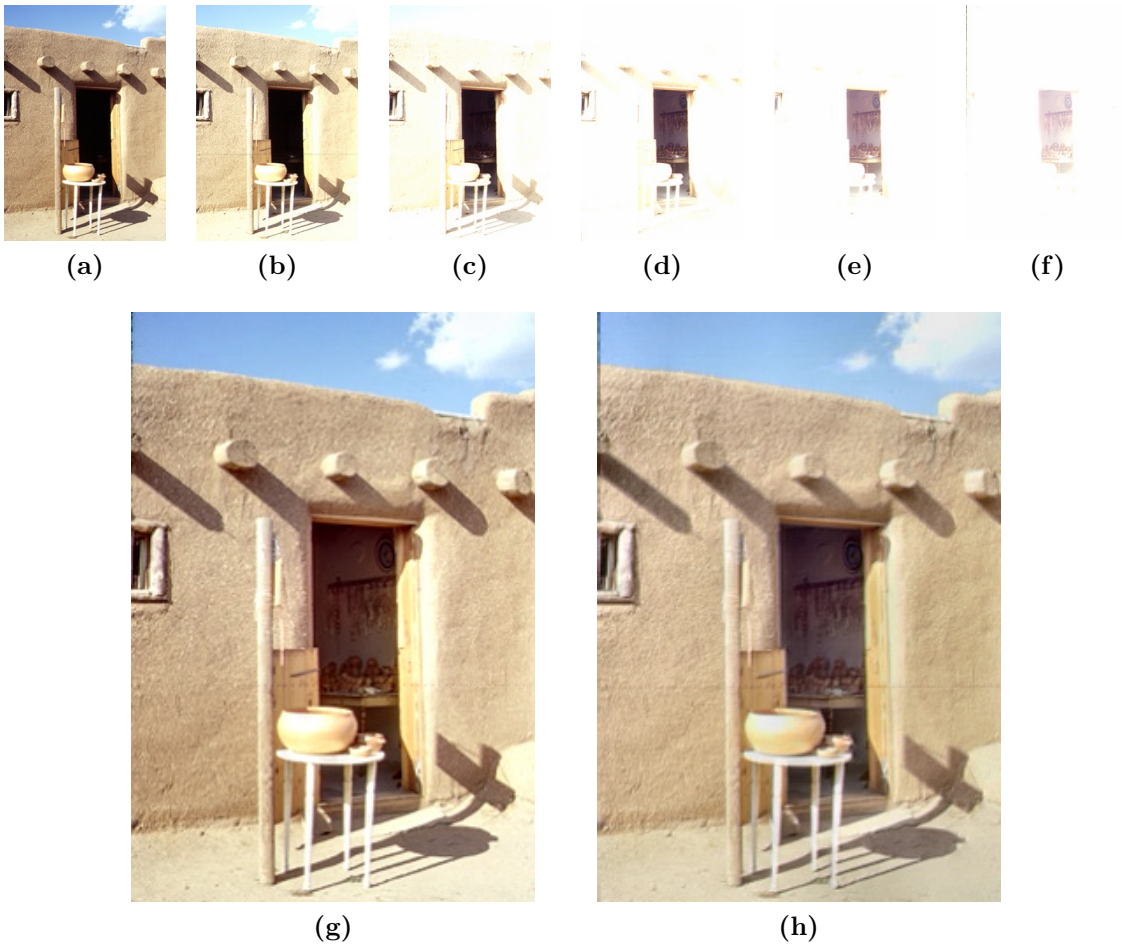


Figure 2.11: The first row contains the input images (**door**). The second row contains the fused image by MEF and proposed algorithm (left to right).

2.7.3.3 Results - multi focus grayscale image fusion

Images **clock** and **pepsi** [41] presented in Figures 2.13a–2.13b and 2.14a–2.14b, are two multi-focus grayscale images used for comparison. The results produced by the proposed fusion algorithm are presented in Figures 2.13e and 2.14e, respectively. For visual comparison, we consider the results using two methods presented in the literature for multi-focus grayscale images, the DCT (Figures 2.13c and 2.14c) and SVD (Figures 2.13d and 2.14d) methods. It may be noted in Figure 2.13f, that the DCT method produces undesirable blocking artifacts. The SVD method also produces artifacts that are more clearly visible in Figure reffig:fig28, on the zoomed in object edges. On the other hand, the proposed algorithm produces a good fusion of the two multi-focus images and is free from visual artifacts.

Figure 2.12: The first row contains the input images (**house**). The second row contains the fused image by MEF and proposed algorithm (left to right).

2.7.3.4 Results - multi focus color image fusion

Figure 2.15 presents an example of multi-focus fusion done with the proposed method for a pair of color images named **foot**. [42]. In the image shown in Figure 2.15a, the foreground is in focus, while in Figure 2.15b, the background region, with the writing, is in focus. The objective is to merge the two images and generate an all in focus image, that looks realistic and has minimal noticeable artifacts. The result shown in Figure 2.15c illustrates that the proposed algorithm can successfully fuse together color images and generate a good visual quality result.

2.7.3.5 Efficiency analysis

In addition to visual comparison, efforts have been made for quantitative comparison using objective metrics. To the best of our knowledge, in literature there exists no objective quality measure to evaluate the results of image fusion techniques. One of the main reasons behind this appears to be the fact that in most frameworks there exists no ideal fused image that can be used as benchmark. This has led researchers to develop metrics like Edge Content (EC) [27], [33] Second Order Entropy (SOE) [33], Blind Image Quality (BIQ) [43] and others. These metrics do not require an ideal fused image for comparison, but are prone to give misleading results in the presence of noise and/or blocking artifacts. For example, EC is an average measure of the gradient magnitudes of an image and methods producing blocking artifacts lead to

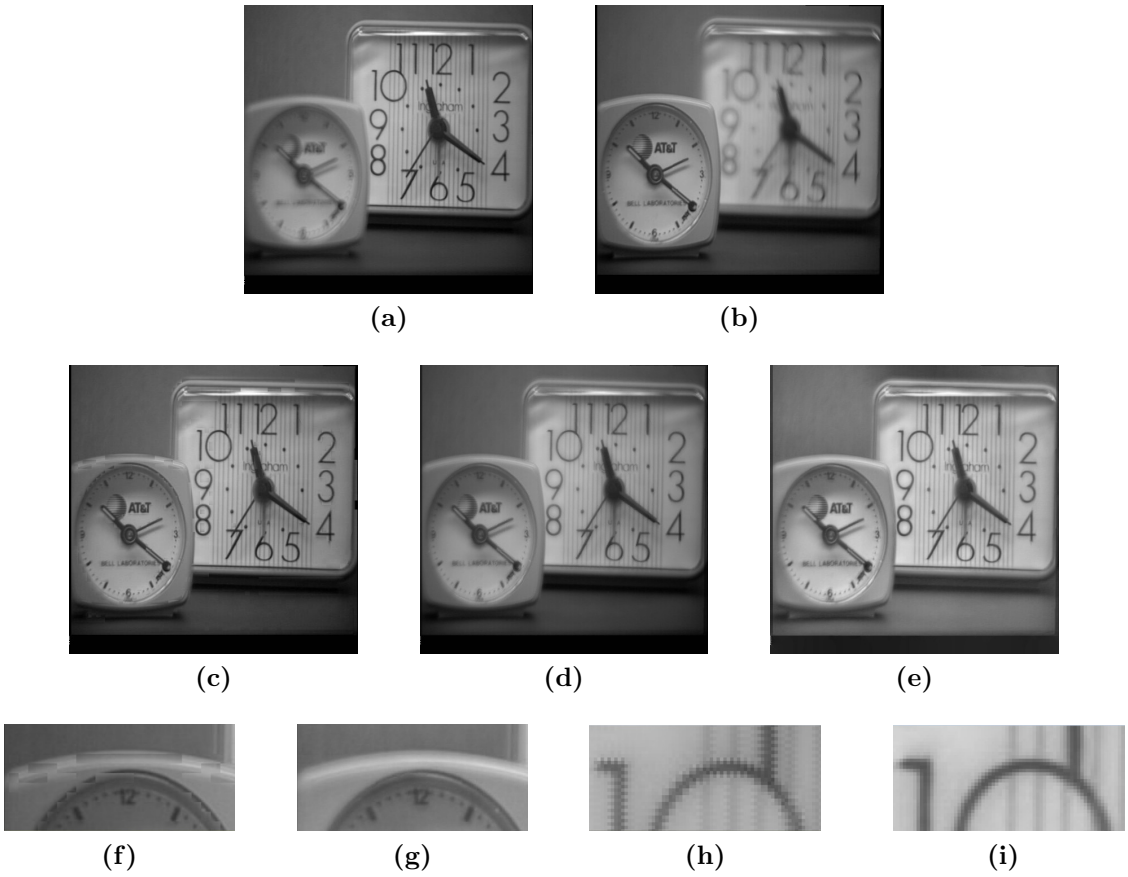


Figure 2.13: Multi focus fusion of grayscale image. The first row contains the input images (**clock**). The second row contains the fused image by DCT, SVD and proposed algorithm (left to right). 2.13f and 2.13h are zoomed in portions of the fused image by DCT and SVD respectively, 2.13g and 2.13i are the corresponding zoomed portions of the image fused by proposed algorithm.

higher EC values. Similar problems are associated with SOE and BIQ, as they are both variations of information and entropy of the image. Thus we have refrained from comparing the results quantitatively using such metrics. Comparison with respect to computational time is presented in Table 2.2 (using Intel R Core TM i3-3110M @ 2.4 GHz and 4GB RAM). It should be noted that the time presented in the table is normalized with respect to the total number of pixels present in the image.

2.8 Conclusions

In this chapter, a new way to reconstruct images from a given gradient data set was presented. The focus of the studies was on non-square input images. The new

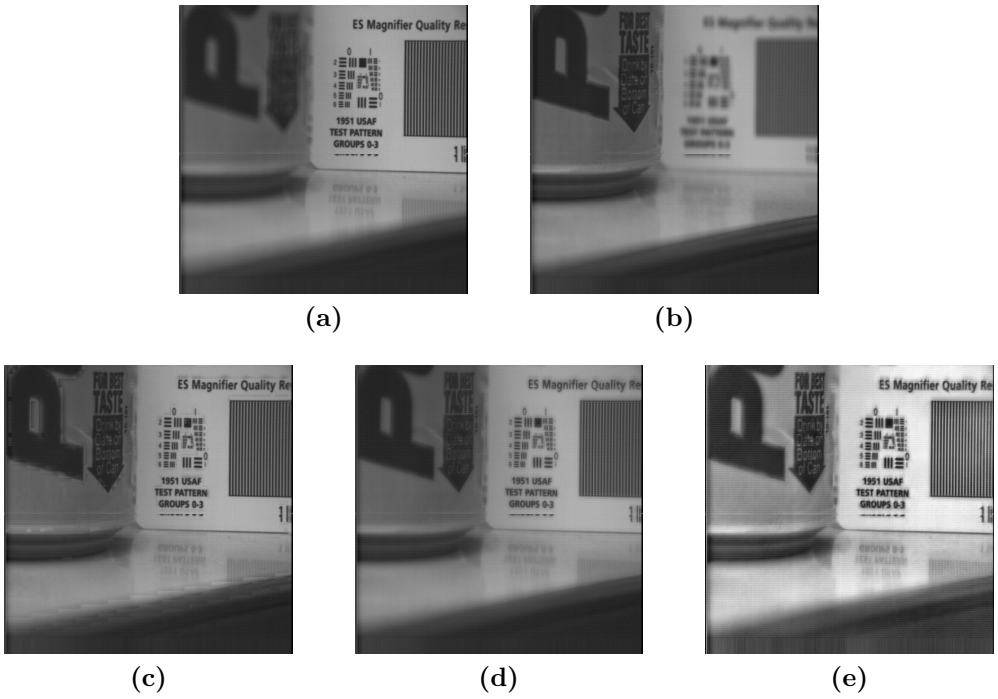


Figure 2.14: The first row contains the input images (**pepsi**). The second row contains the fused images produced by DCT, SVD and proposed algorithm (left to right).

Figure 2.15: Multi focus fusion of color images. The input images and the result obtained using the proposed algorithm are shown left to right.

reconstruction approach developed in this chapter provided an opportunity for a closer look on the effect of errors in the approximation subband coefficients of the wavelet decomposition of an image. The effect of these errors was illustrated on an example, and this helped develop a more efficient way to reconstruct a non-square image from gradient data. The algorithm was then included in a new image fusion technique, that is used to fuse together multi-exposure and multi-focus images, in color or grayscale representation. The insight gained in this chapter opens the gate

Image name	DCT	SVD	MEF	GrW	Proposed method
Clock	0.0780	0.0556			0.0224
Pepsi	0.0707	0.0573			0.0256
Foot					0.0134
Door			0.0650		0.0305
House			0.0296		0.0248
Igloo				0.2798	0.0714
Monument				0.1469	0.0340

Table 2.2: Average computational time per pixel ($\times 10^{-4}$)

for more efficient ways of reconstructing non-cubic signals from gradient data, with applications in video processing, and this will be explored at large throughout the next chapter of this dissertation.

Chapter 3

Three dimensional signal reconstruction from gradient data

3.1 Chapter outline

A wavelet based algorithm for reconstructing three dimensional (3-D) signals from gradients is proposed in this chapter. The algorithm is based on obtaining the Haar wavelet decomposition of the signal from the given gradient data set, and from it the 3-D signal using wavelet synthesis, with the possibility of including an iterative Poisson solver at each resolution level.

The motivation behind the developments of this chapter is given in Section 3.2. An overview of the signal of interest and several targeted applications are in Section 3.3. Section 3.4 introduces the notation and Section 3.5 presents the details of an algorithm developed to recover a video sequence from a given gradient data set. The chapter continues with an analysis aimed at evaluating the performance of the proposed technique to recover 3-D signals from their derivatives, presented in Section 3.6. In Section 3.7, the 3-D reconstruction technique is included in two gradient domain based video editing applications. In Section 3.8, the content of the chapter is briefly reviewed and the main conclusions are drawn.

3.2 Motivation for 3-D study

Efficient algorithms to reconstruct 3-D surfaces are important in applications from a variety of fields, such as computer graphics, robotics, medicine, or film industry. A

brief overview of the most common applications involving 3-D surface reconstruction studied in these fields is given below, with the purpose of motivating our interest for developing a 3-D surface reconstruction algorithm.

Computer graphics study 3-D surface reconstruction for applications such as documenting historical sites or works of art. One of the first such endeavors is The Digital Michelangelo Project [44], where a team of researchers scanned works of Michelangelo with the goal of producing 3-D models that would then be made available worldwide.

For many medical conditions, medical image analysis is a critical step on the way to establish a patient's diagnosis and to determine the optimal treatment plan. However, sensors used in medical imaging have limitations, and this makes algorithms that generate accurate reconstructions of the 3-D surface of the analyzed body part, become a necessary an important tool.

The work developed in this chapter targets applications from film industry, and devises an algorithm that reconstructs a digital video from gradient data. The algorithm, once devised, will be used in gradient based video editing techniques, and this will be illustrated in Section 3.7 of this chapter.

3.3 Related work

Researchers from the computer graphics [45], [46] and medical imaging [47], [48], [49] communities study the problem of 3-D surface reconstruction form a slightly different perspective than the one adopted in this work. Specifically, they reconstruct a 3-D surface from point clouds, with or without orientations (i.e. surface normals, or gradients). In our work, we reconstruct a 3-D digital signal from gradient data, and the average value of the signal to be reconstructed.

As argued earlier, obtaining a digital signal from a given gradient data set is an important required step in gradient domain signal processing applications. This step poses a problem when the gradient from which the reconstruction is attempted is artificially generated, and does not belong to a digital signal.

While many techniques have been developed to reconstruct 2-D surfaces from a given gradient dataset [11], [50], [51], [52], the 3-D case appears less explored. The most used solution [9] to solve the 3-D problem of signal reconstruction from a gradient dataset belongs to the class of multigrid [53], [14]. Multigrid techniques are a class of highly efficient solvers for the Poisson problem, based on solving the problem on coarse grids, followed by interpolating a correction term back to finer

grids. Practical multigrid algorithm use equal spacing along all dimensions at any given scale, imposing thus an implicit restriction on the signal size, namely to have all dimensions equal, and power-of-two. In this chapter, a wavelet based algorithm for reconstructing 3-D signals from gradients is proposed. The proposed technique does not require that all signal dimensions be equal, and can be used to reconstruct 3-D signals such as digital videos, which tend to have all three dimensions different (width and height of frame, and number of frames). The algorithm is based on obtaining the Haar wavelet decomposition of the signal directly from the gradients and from it the 3-D signal using wavelet synthesis, with the possibility of including an iterative Poisson solver at each resolution level.

3.4 Notation

The technique developed in this chapter targets applications from film industry, such as video editing, and as such, the signals of interest are video signals. That is, the signal of interest is a 3-D discrete signal in which every point in space and in time is described by a brightness value. Unless otherwise specified, the video signals are considered in grayscale representation.

Let Φ be a 3-D signal, with size $2^M \times 2^N \times 2^P$, where $1 < M \leq N \leq P$ are integer

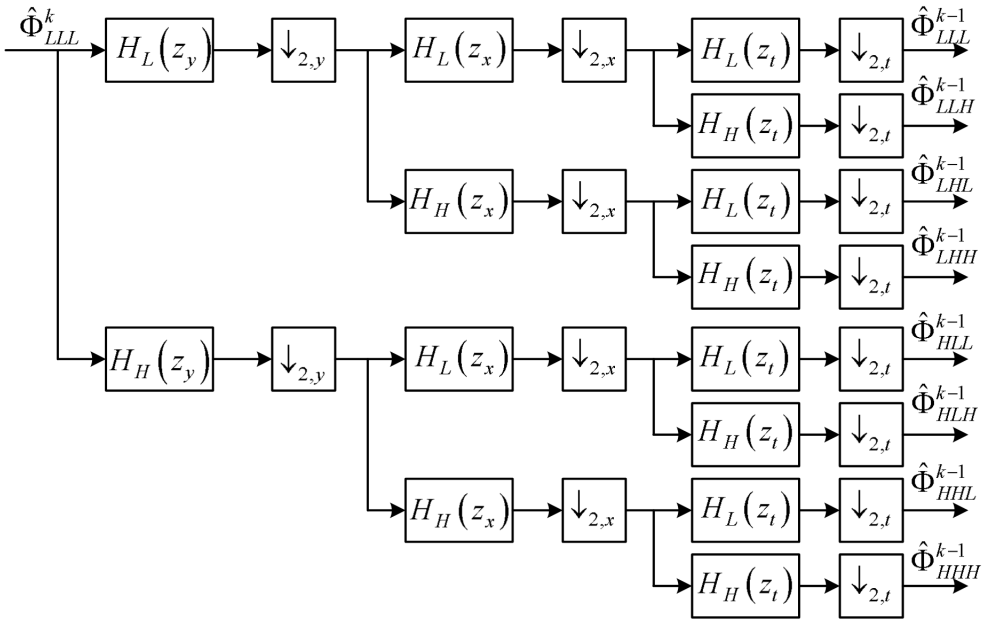


Figure 3.1: One step in the wavelet decomposition of a 3-D signal

numbers. The signal value at a point (y, x, t) is denoted $\Phi(y, x, t)$, with y, x and t orthogonal directions.

Directional downsampling (subsampling) by an integer factor n along the x, y or t direction is denoted $\downarrow_{n,x}$, $\downarrow_{n,y}$ or $\downarrow_{n,t}$, respectively, while directional upsampling assumes a similar notation, with the help of the $\uparrow$ symbol.

When the downsampling or upsampling operations are performed along all directions, and there is no danger of confusion, the subscript that indicates the direction is omitted. For example, downsampling a 3-D signal Φ by a factor of 2 along directions x, y and t is denoted $\downarrow_2 \Phi$.

The Haar wavelets are relevant to the algorithm developed here and the key building block in obtaining the wavelet decomposition of a 3-D signal is briefly reviewed here.

The separability property of the 3-D wavelet transform allows for the Haar wavelet decomposition of a 3-D signal Φ to be obtained after M steps illustrated in Figure 3.1. The convention is that when $k = M$ the $\hat{\Phi}^k_{LLL}$ signal is the original signal, with size $2^M \times 2^N \times 2^P$. For the definition of the Haar wavelet filters the reader is referred to Equations 2.1–2.2 in Chapter 2.

3.5 Three dimensional signal reconstruction from gradient

The wavelet-based reconstruction technique presented in Chapter 2 is developed here in the case of 3-D signals, without expanding the data size to nearest cube, by finding an approximation of the Haar wavelet decomposition of the signal directly from the gradients and from it the 3-D signal using wavelet synthesis. As such, the result is obtained in two steps, called analysis and synthesis, and their details are presented over the following sections.

3.5.1 Analysis step: detail subbands of the wavelet decomposition

The objective of the analysis step is to produce an approximation of the Haar wavelet decomposition of the signal of interest from a given gradient data set. The eight subbands of the first level in the Haar wavelet decomposition of a 3-D signal are

shown in Figure 3.2. The problem that we address here is how to find these subbands from the three directional components of the given gradient of the unknown 3-D signal.

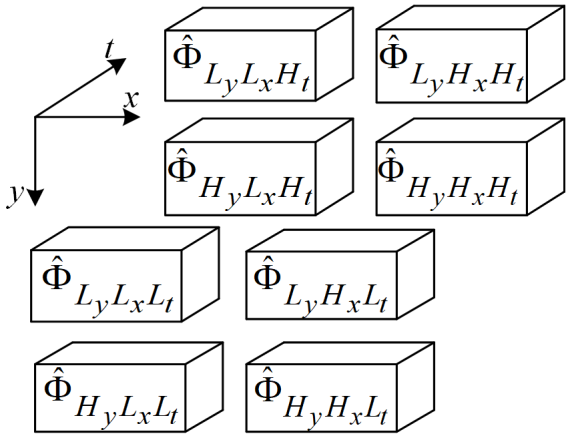


Figure 3.2: First eight subbands in the waveet decomposition of a 3-D signal

Let Φ_x, Φ_y and Φ_t be the three directional components of the given gradient data, from which the signal Φ is to be reconstructed. In the beginning of the analysis step, the highest resolution subbands HLL, LHL and LLH of the 3-D Haar wavelet decomposition of the unknown signal Φ are obtained from the given gradient components, by lowpass filtering and subsampling by 2:

$$\hat{\Phi}_{HLL}^{M-1} = \frac{1}{\sqrt{2}} \downarrow_2 \{\Phi_x H_L(z_y) H_L(z_t)\} \tag{3.1}$$

$$\hat{\Phi}_{LHL}^{M-1} = \frac{1}{\sqrt{2}} \downarrow_2 \{\Phi_y H_L(z_x) H_L(z_t)\} \tag{3.2}$$

$$\hat{\Phi}_{LLH}^{M-1} = \frac{1}{\sqrt{2}} \downarrow_2 \{\Phi_t H_L(z_x) H_L(z_y)\} \tag{3.3}$$

In Equations 3.1–3.3, the expressions $\Phi_x H_L(z_y)$, $\Phi_y H_L(z_x)$ and $\Phi_t H_L(z_x)$ denote filtering the signals Φ_x, Φ_y and Φ_t with the Haar lowpass analysis filter, along the y, x, and y directions, respectively. The expression $\Phi_x H_L(z_y) H_L(z_t)$ denotes successive filtering of the signal Φ_x with the Haar lowpass analysis filter, first along the y direction, then along the t direction.

Next, the HHH subband is found as as a linear combination of the given gradient components, highpass filtered along orthogonal directions and downsampled by 2:

$$\hat{\Phi}_{HHH}^{M-1} = \frac{1}{6\sqrt{2}} \downarrow_2 \left\{ \Phi_x H_H\left(z_y\right) H_H\left(z_t\right) + \Phi_y H_H\left(z_x\right) H_H\left(z_t\right) + \Phi_t H_H\left(z_x\right) H_H\left(z_y\right) \right\} \tag{3.4}$$

The HHL, LHH and HLH subbands of the Haar wavelet decomposition (shown in Figure 3.2) are also obtained from the given gradient components by successive highpass and lowpass filtering, and subsampling by 2:

$$\hat{\Phi}_{HHL}^{M-1} = \frac{1}{\sqrt{2}} \downarrow_2 \left\{ \Phi_x H_H\left(z_y\right) H_L\left(z_t\right) \right\} \tag{3.5}$$

$$\hat{\Phi}_{LHH}^{M-1} = \frac{1}{\sqrt{2}} \downarrow_2 \left\{ \Phi_y H_H\left(z_t\right) H_L\left(z_x\right) \right\} \tag{3.6}$$

$$\hat{\Phi}_{HLH}^{M-1} = \frac{1}{\sqrt{2}} \downarrow_2 \left\{ \Phi_t H_H\left(z_x\right) H_L\left(z_y\right) \right\} \tag{3.7}$$

The final subband in the decomposition shown in Figure 3.2, i.e. $\hat{\Phi}_{LLL}^{M-1}$, is obtained by repeating the process described by Equations 3.1–3.7 at lower resolution levels. In this way, the detail subbands at all resolution levels are found.

3.5.2 Analysis step: approximation subband at lowest resolution

After the process described in Section 3.5.1 is completed, the lowest resolution LLL subband (i.e. $\hat{\Phi}_{LLL}^{0}$) needs to be found in order to obtain the complete wavelet decomposition of the unknown signal Φ.

When the signal is cubic, i.e. when the size of the unknown signal is $2^M \times 2^M \times 2^M$, the lowest resolution subband is a scalar. The value of this scalar cannot be obtained from the given gradient components, but can be estimated from the mean value of the signal. Specifically, $\hat{\Phi}_{LLL}^{0} = \left(2\sqrt{2}\right)^M \cdot m$, where m is the mean value of the unknown signal, or an estimate thereof.

When the signal is not cubic, the lowest resolution subband of the Haar wavelet decomposition of Φ is denoted $\hat{\Phi}_{L_y L_x L_t}^{0}$ and is a signal with size $1 \times 2^{N-M} \times 2^{P-M}$.

Obtaining this subband is a 3-D generalization of the procedure described in Chapter 2, Section 2.5.2, and illustrated by the numerical example included in Appendix A. Specifically, $\hat{\Phi}_{L_y L_x L_t}^{0}$ can be found by solving a system of linear equations, obtained

from the given discrete partial derivatives and the mean value of the signal. The partial derivative Φ_x is used to derive $2^{P-M}\left(2^{N-1}-1\right)$ equations, and the partial derivative Φ_t is used to derive $2^{N-M}\left(2^{P-1}-1\right)$ equations. The mean value of the signal yields one more equation. In this way, a system of linear equations is formed:

$$A\lambda = B \tag{3.8}$$

where:
$A \in \mathbb{R}^{\left[2^{P-M}\left(2^{N-1}-1\right)+2^{N-M}\left(2^{P-1}-1\right)+1\right]\times 2^{N+P-2M}}$, and $B \in \mathbb{R}^{\left[2^{P-M}\left(2^{N-1}-1\right)+2^{N-M}\left(2^{P-1}-1\right)+1\right]\times 1}$ are known quantities obtained from partial derivatives and the mean value of the signal, and λ is the unknown, which, once found is reshaped and generates $\hat{\Phi}^0_{L_yL_xL_t}$.

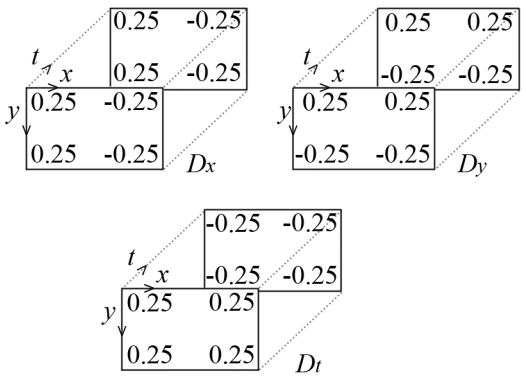


Figure 3.3: The 3-D arrays D_x, D_y and D_t used in Eq. 3.9

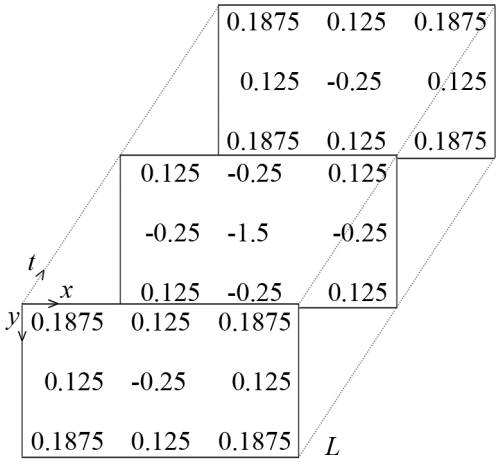


Figure 3.4: The 3-D array L used in Eq. 3.9

3.5.3 Synthesis step

In the synthesis step, Haar wavelet synthesis is used to obtain the 3-D signal of interest Φ. An iterative Poisson solver is included at each resolution level, to alleviate the effects of approximating the coarsest resolution subband coefficients. Incorporating the Poisson solver in the synthesis step at resolution level k is done as follows: first, Haar wavelet synthesis is performed at level k of the decomposition obtained in the analysis step, and resolution level $k+1$ is obtained; then, a 3-D iterative Poisson solver is applied to the $k+1$ resolution level. The solver is a 3-D extension of the one used in [54]. Its role is to smooth out any errors in the decomposition obtained at the end of the analysis step. The first level where the Poisson solver can be applied is the second and the last one is the maximum resolution level of the considered signal. The equation describing the 3-D Poisson solver employed in the synthesis step is:

$$\hat{\Phi}^k_{LLL}(i+1) = \hat{\Phi}^k_{LLL}(i) + 0.25\left[\ L \otimes \hat{\Phi}^k_{LLL}(i) - D_x \otimes \tilde{\Phi}_x - \ D_y \otimes \tilde{\Phi}_y - D_t \otimes \tilde{\Phi}_t\right]$$

where $i \geq 1$ is the integer iteration index, $\tilde{\Phi}_x$, $\tilde{\Phi}_y$ and $\tilde{\Phi}_t$ are subsampled and filtered versions of the gradient components and are obtained in the analysis step, $\hat{\Phi}^k_{LLL}(1)$ is the LLL subband at resolution level k, D_x, D_y and D_t are listed in Figure 3.3, L in Figure 3.4, and $\otimes$ denotes 3-D convolution. The use of the wavelet transform implies a computational complexity of $\mathcal{O}(N)$ and since in most cases a very low number of iterations of the Poisson solver are required at each resolution level, including the Poisson solver in the synthesis step does not have a significant impact on the computational complexity of the proposed algorithm.

3.6 Performance evaluation

In this section, the reconstruction algorithm is evaluated and the quality of the results is compared with existing techniques. The comparison is done first with respect to an earlier version of the algorithm [16], which requires signal extension to the same nearest power of two along each of the signal dimensions. Then, the proposed technique is compared with a fast and efficient multigrid technique [14], used in the video processing community [9], on two video sequences with different size.

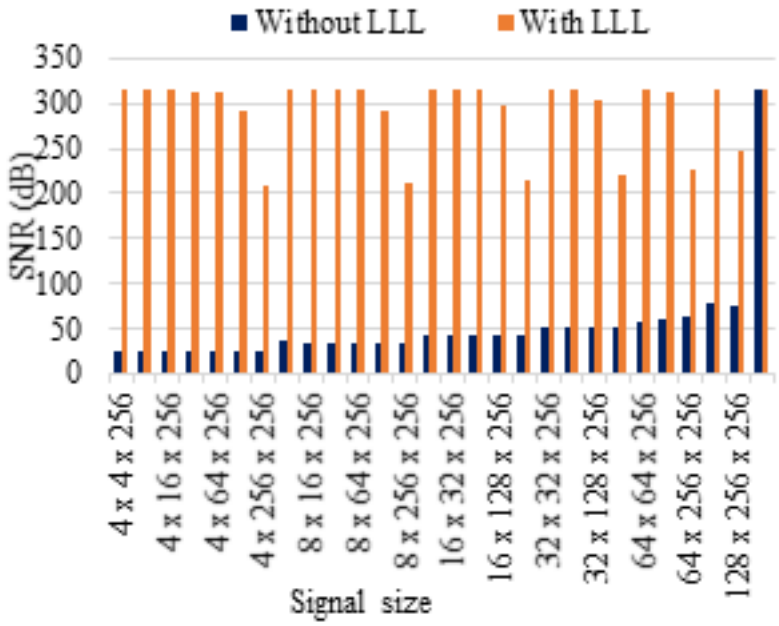


Figure 3.5: Performance evaluation in terms of solution accuracy

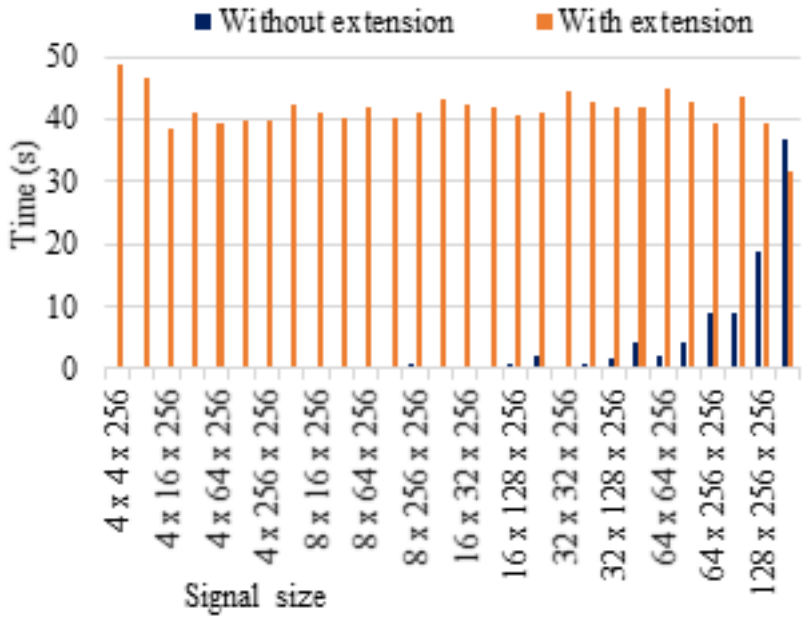


Figure 3.6: Performance evaluation in terms of solution efficiency

3.6.1 Comparison with previous work

A set of 3-D test signals with sizes ascending powers of two was generated. The gradient of each test signal was computed and reconstructions obtained by computing

the coarsest resolution subband using the method in Section 3.5 (denoted "With LLL" in Figure 3.5) were compared with reconstructions where the coarsest resolution subband has all entries equal to a value proportional to the signal mean (denoted "Without LLL" in Figure 3.5). Three iterations of the recursive Poisson solver were included at each resolution during the synthesis algorithm. In terms of solution accuracy (SNR) the quality of the results is correlated with the ones reported in the 2-D case, in Section 2.6.

The significant advantage of the new reconstruction technique over the approach that performs signal extension, is in terms of efficiency. The reconstructions obtained by setting all entries in $\hat{\Phi}^0_{LLL}$ to a constant are compared to the earlier technique in [10], where a cubic wavelet decomposition of the signal extension was obtained in the analysis step, from an extended version of the given gradient. The results are shown in Figure 3.6. As the Figure reveals, the new algorithm outperforms the previous approach in terms of speed especially in cases where the differences in the signal size are large.

3.6.2 Comparison with multigrid

Here, the performance of the technique developed in Section 3.5 to reconstruct a 3-D signal from a given gradient dataset is assessed by comparing it with an existing technique used by the video processing community [9]. The performance evaluation was done by generating two testing video sequences with different sizes, and comparing the reconstructions produced by the proposed algorithm with those produced by the fast multigrid algorithm developed by Prof. A. J. Roberts [14].

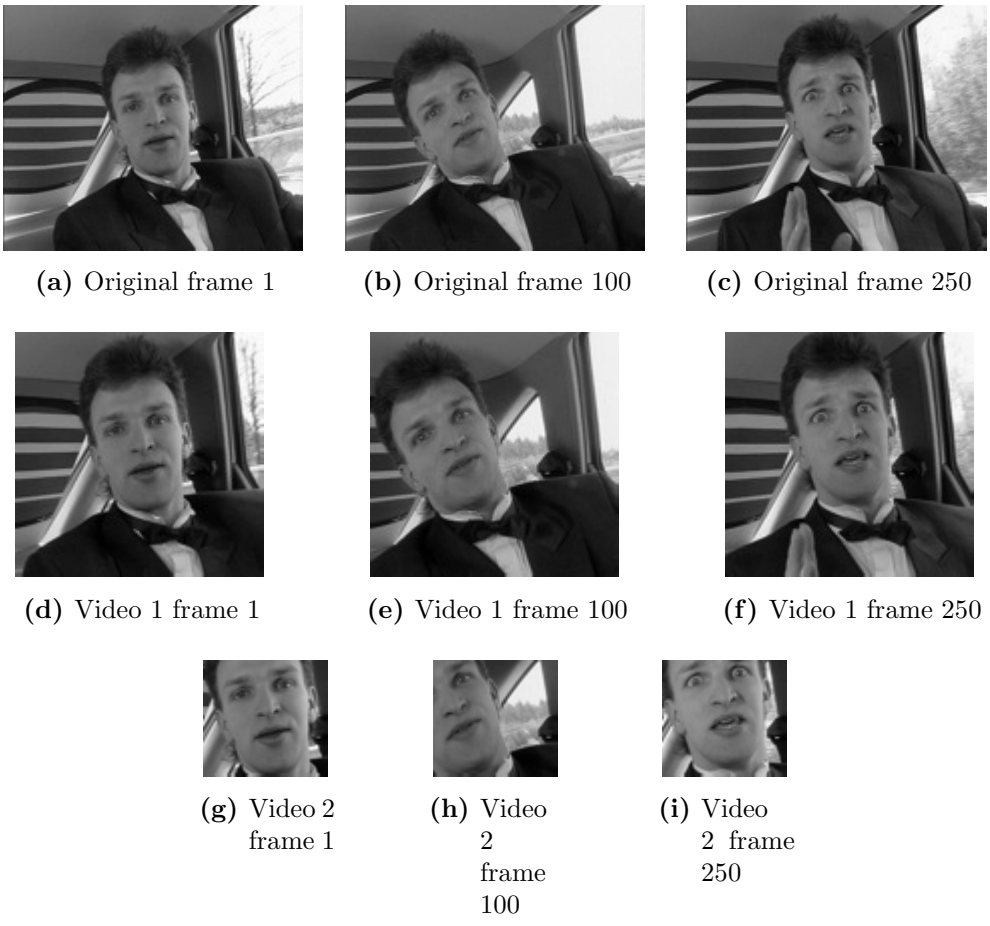
The video sequence used in this experiment is called **carphone** and is one of the typically employed benchmark video sequences in video processing. Several sample frames from the original video are shown in Figures 3.7a– 3.7c. The size of the original video sequence is $144 \times 176 \times 382$. Two testing video sequences were created by cropping each frame in the original video: Video 1, with size $122 \times 124 \times 382$, and Video 2, with size $58 \times 62 \times 382$.

The following experiments were carried out on Video 1 and Video 2. The gradient was computed, and the average value of the video sequence was calculated and stored. Then, the proposed reconstruction technique was used to recover the video sequence, from the gradient data set. The final coarse resolution subband entry was obtained exactly, as in the procedure described in Section 3.5.2. The available implementation

of the multigrid technique assumes the domain of the function to be recovered is cubic. So, the two 3-D test signals were first extended to a cube version, then their gradient was computed and used as input to the reconstruction algorithm. Three iterations of Poisson solvers were used in each of the algorithms, for a fair comparison.

The recovered video sequences were compared with the originals and the following four performance indices were monitored:

- **SNR**: the signal to noise ratio between vectorized versions of the original and reconstructed signals
- **PSNR**: the peak signal to noise ratio between each original and each reconstructed frame; the average value across all frames was recorded



(a) Original frame 1 **(b)** Original frame 100 **(c)** Original frame 250

(d) Video 1 frame 1 **(e)** Video 1 frame 100 **(f)** Video 1 frame 250

(g) Video 2 frame 1 **(h)** Video 2 frame 100 **(i)** Video 2 frame 250

Figure 3.7: Sample frames from **carphone** video sequence. Size of original video: $144 \times 176 \times 382$. Size of Video 1: $122 \times 124 \times 382$. Size of Video 2: $58 \times 62 \times 382$. Original video from [55].

	Proposed	Multigrid [14]
Overall SNR	79	47
Average PSNR	87	55
Average SSIM	1	0.9997
Time per sample	2.18×10^{-5}	4.39×10^{-5}

Table 3.1: Performance evaluation on Video 1

	Proposed	Multigrid [14]
Overall SNR	91	48
Average PSNR	98	54
Average SSIM	1	1
Time per sample	2.28×10^{-5}	15.2×10^{-5}

Table 3.2: Performance evaluation on Video 2

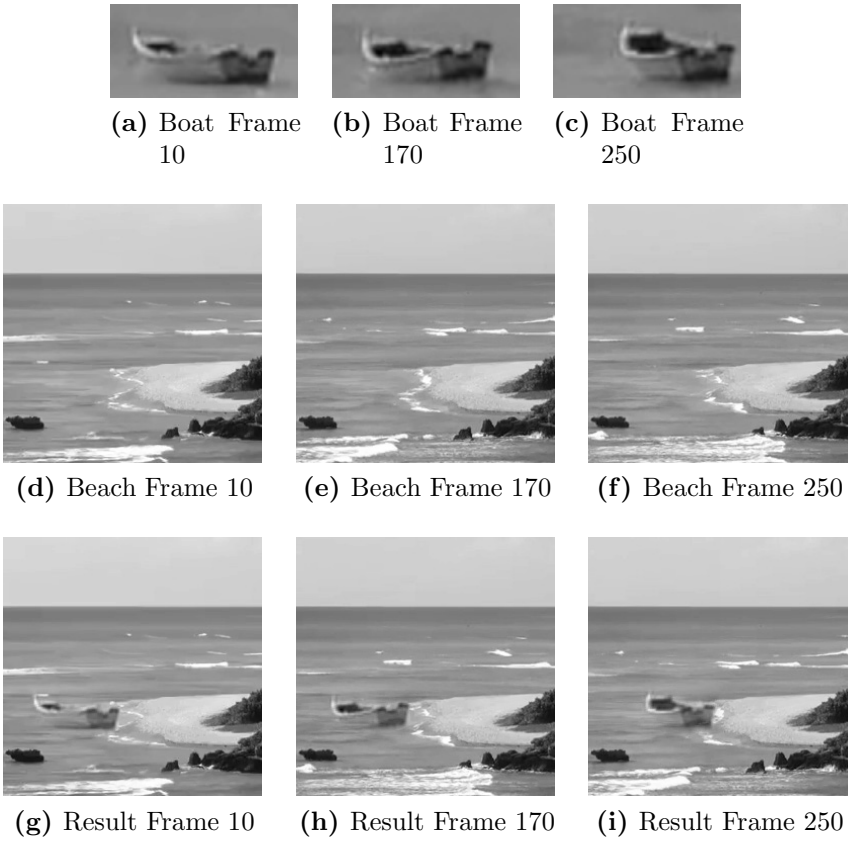
- **SSIM**: the structural similarity index measure [17] between each original and each reconstructed frame; the average value across all frames was recorded
- **time per sample**: the time necessary to generate the gradient data set and obtain an approximation of the reconstruction was measured, and the result divided by the number of samples in the original 3-D signal

The results pertaining to the experiments performed on Video 1 and Video 2 are shown in Table 3.1 and Table 3.2. In terms of visual quality (SNR, PSNR and SSIM), the results produced by the two techniques are very similar. The advantage of the proposed technique is in that it can handle signals with large difference between the three sizes in a more efficient way, and this becomes apparent from Table 3.2.

3.6.3 Conclusions regarding 3-D performance

The results shown in Figures 3.5 and 3.6, as well as those in Table 3.1 and 3.2 show that for 3-D signals with large differences between the three dimensions, the new approach produces a solution in significantly less time than the approach in [10]. This is expected, as a significant amount of memory is required by the algorithm in [10], and [14] to generate and store the extended cubic intermediate signals, and this translates into additional processing time or exhausting system memory.

Moreover, as at the end of the analysis step the algorithm produces a wavelet-like decomposition of the 3-D signal of interest, additional processing such as denoising becomes possible. This is a unique feature that distinguishes the proposed algorithm



(a) Boat Frame 10 **(b)** Boat Frame 170 **(c)** Boat Frame 250

(d) Beach Frame 10 **(e)** Beach Frame 170 **(f)** Beach Frame 250

(g) Result Frame 10 **(h)** Result Frame 170 **(i)** Result Frame 250

Figure 3.8: Beach - boat example

from existing state of the art reconstruction algorithms and makes it valuable for gradient based video processing applications.

3.7 Applications

In this Section, two video editing applications will be presented. In the first one, gradient information from two video sequences will be combined with the purpose of generating a new one, that contains information from both video sequences. In the new video, content from a user specified region will be replaced with content from the other one. In the second application, gradient information from two videos will be combined with a slightly different objective. Specifically, the objective in this application is to generate a third video that simulates a transparency effect. In both applications, a requirement for the resulting video sequence is that it has to be realistic, and free as much as possible from any noticeable visual artifacts.

3.7.1 Video editing: content replacement

3.7.1.1 Example 1: Beach boat (grayscale)

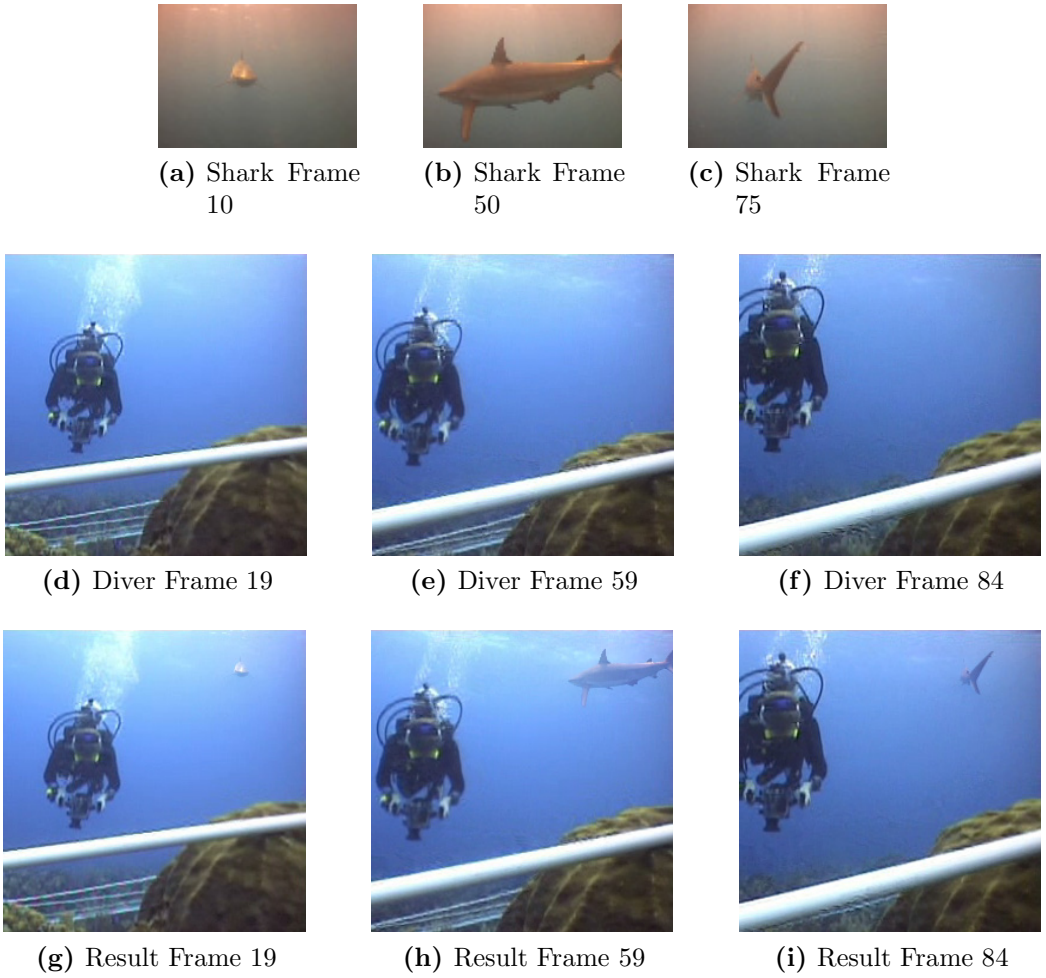
This example illustrates the framework of our technique in its most basic form. The two videos to be combined are in grayscale representation and are illustrated by three representative frames in Figure 3.8g–3.8f.

The objective is to insert the content of the `boat` video in the `beach` video sequence [1]. Our approach to the task is the following. First, the gradient of the two video sequences is computed. Then, a spatial location of where the boat video is to be inserted is specified by the user. Instead of stitching along the edge of the `boat` video, we do the stitching in regions where the magnitude of the spatial gradient of the `boat` video exceeds the value of the sum between the mean and median values of the spatial magnitude of the `boat` video gradient. This spatial gradient thresholding intuitively corresponds to edge detection and the value of the threshold was chosen to minimize user interaction. Within the user specified spatial boundaries, we use the magnitude of the spatial gradient of the `boat` video to create a mask, with values one in regions where it is higher than the threshold and zero everywhere else. Then, we use this mask to select the corresponding components from the gradient of the `boat` video and place them in the gradient of the `beach` video. Next, a composite video is reconstructed from the gradient data thus obtained using the algorithm in section 3.5. Lastly, the mean value of the video is corrected to match the mean value of the `beach` video. Three representative frames of the result are shown in Figure 3.8g–3.8i.

3.7.1.2 Example 2: Diver - Shark (color)

This example illustrates the framework of our technique for color videos. The two videos used here are illustrated by three representative frames in Figure 3.9a– 3.9f. The objective is to include the content of the `shark` video sequence in the `diver` video sequence. Combining the gradients is done on each of the three color channels individually. The three stitched gradient components (corresponding to the red, green and blue channels) are obtained in a similar manner as described in the previous example. Then, the intensity values of the red, green and blue channels of the composite video are obtained using the proposed reconstruction algorithm. At the end, the three channels are combined to produce the color video. Three frames of

[1] The videos are from YouTube



(a) Shark Frame 10 **(b)** Shark Frame 50 **(c)** Shark Frame 75

(d) Diver Frame 19 **(e)** Diver Frame 59 **(f)** Diver Frame 84

(g) Result Frame 19 **(h)** Result Frame 59 **(i)** Result Frame 84

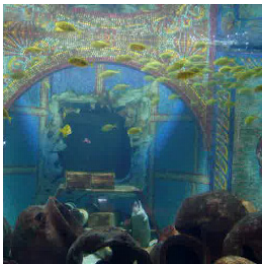
Figure 3.9: Diver - shark example

the result are shown in Figure 3.9g–3.9i. This example differs from the previous one in several ways. First, the `shark` video sequence has fewer frames than the diver sequence. Still, the transition from one video sequence to the other is smooth in both space and time, and no seams are noticeable. Second, the two videos have strong differences in colour, as can be noted from Figure 3.9g–3.9i. Although the dominant color is different in the two video sequences, the result looks uniform. This is because after the reconstruction, the mean value of the result was adjusted to match the one of the dominant video (in this case, the `diver` video).

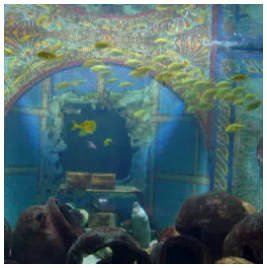
(a) Clock Frame 5

(b) Clock Frame 19

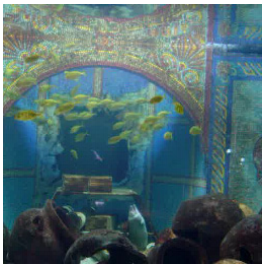
(c) Clock Frame 50



(d) Fish Frame 5



(e) Fish Frame 19



(f) Fish Frame 50

(g) Result Frame 5

(h) Result Frame 19

(i) Result Frame 50

(j) Result from [9] Frame 5

(k) Result from [9] Frame 19

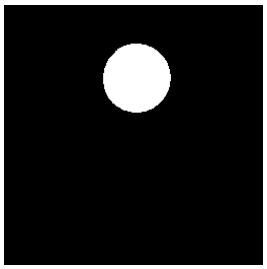
(l) Result from [9] Frame 50

Figure 3.10: Clock - fish example

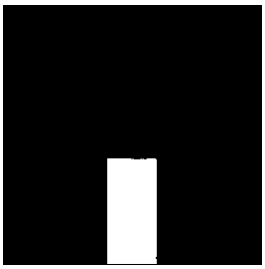
3.7.2 Video editing: transparency

This example was studied in [9] and is presented here to showcase that the proposed reconstruction technique is also a valuable tool in transparency video editing applications.

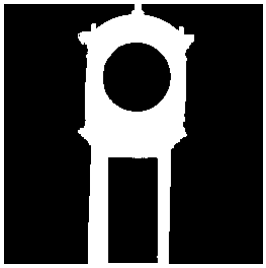
(a) Clock video frame



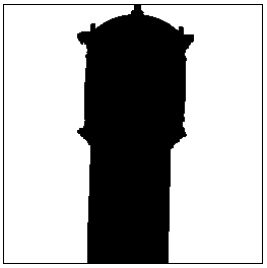
(b) Clock Face



(c) Pendulum Face



(d) Clock Rim



(e) Outside Clock

Figure 3.11: Clock - fish example - masks

The two videos used here [2] are illustrated by three representative frames on the top two rows of Figure 3.10. The two videos have the same spatial size (i.e. the same number or rows and columns in a frame), and differ in the third dimension (i.e., have different number of frames). The objective is to insert the content from the clock video in the fish video and to make the clock and pendulum faces look transparent such that content from the fish sequence appears superimposed. The approach of [9] for this example is gradient based, similar to ours. What is different in our work is the generation of the mask used for the stitching and the reconstruction of the video from the gradient. In an attempt to minimize user interaction (as done in [9]), in this work, the edges of the regions of interest are identified by thresholding the intensity values of the RGB channels of the clock video in each frame. However, this still required prior knowledge about the signal content. A simple thresholding of the magnitude of the gradient on one of the channels failed to yield good results, because of the low contrast between the pendulum and the surrounding frame. The four masks used in this example are shown in Figure 3.11. The result is illustrated by the three frames shown in Figures 3.10g–3.10i, side by side with the results from [9].

[2]The videos have been used in [9] and are courtesy of Hongcheng Wang

3.8 Conclusions

An efficient way of reconstructing 3-D signals from gradient data was presented in this chapter. The approach is based on finding the Haar wavelet decomposition of the signal first, then reconstructing the signal by wavelet synthesis. The main advantage of this technique over a previously devised technique [10] is the way in which the case of non-cubic signals is addressed. Resources are allocated to determining the coarsest resolution coefficients of the Haar wavelet decomposition of the unknown signal, in the general non-cubic case. The experiments presented in Section 3.6 show that when there is no noise in the input data, obtaining an accurate approximation of the coarsest resolution coefficients typically increases the accuracy of the solution, to the expense of computation time and memory consumption. On the flipside, when the gradient data has been modified to meet certain requirements, such as those arising in video editing applications, approximating these coefficients has little to no noticeable consequence on the visual quality of the signal.

The findings of this chapter confirm the initial conclusions of Chapter 2, in that the new approach to signal reconstruction from gradient data seems suitable for multi-dimensional signal processing applications (e.g., 4-D signals such as light fields). These ideas will be further developed in Chapters 4 and 5 of this dissertation.

Chapter 4

Four dimensional signal reconstruction from gradient data

4.1 Chapter outline

An algorithm is described here to generate a four dimensional signal from a known gradient data set. The technique targets light field applications and is developed as such.

The motivation behind the developments of this chapter is given in Section 4.2. Section 4.3 provides an overview of the signal of interest and reviews some of its applications. Section 4.4 introduces the notation used in the remainder of this chapter. In Section 4.5, the details of a technique designed to recover a light field from a given gradient data set are revealed. The chapter continues with an analysis aimed at establishing the performance of the proposed technique to recover light fields from their derivatives, presented in Section 4.6. Section 4.7 reviews the content of this chapter providing concluding remarks and recommendations.

4.2 Objective and motivation for 4-D study

The developments of this chapter are motivated to a great extent by the excellent problem solving capability of gradient based approaches in 2-D or 3-D signal processing applications. As such, a direction that will be explored in the remainder of this dissertation is developing gradient domain based solutions for 4-D signal processing applications.

A necessary and important step in developing algorithms that operate primarily in the gradient domain is devising a way to recover the signal from a given gradient data set, and this is the main goal of this chapter. The significant contribution of this work is developing an efficient algorithm to reconstruct a light field from a given gradient datasets.

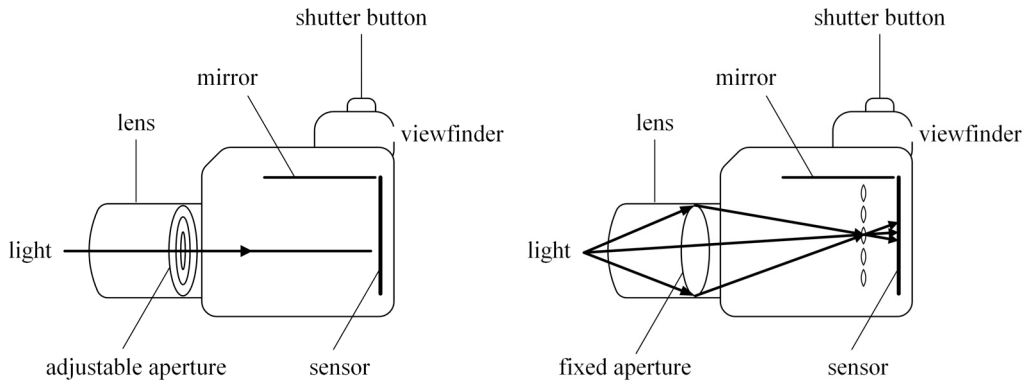
4.3 Background information

4.3.1 Light field scene representation

Recent developments [56] put forth new ways of representing light in a scene. In this novel approach, light in the scene is no longer represented as a 2-D function (i.e. brightness in the scene as a function of its position on a plane). Instead, the light intensity of a point in a scene is depicted as a 4-D function, that depends on the spatial position of the point in a 2-D plane and on the direction that a certain ray of light that hits that point comes from.

Obtaining such a scene representation can be tackled in different ways. For example, an array of cameras can be used to capture multiple representations of the same scene [57], [58]. Another way is to use one camera, translate it on a regularly spaced grid to generate an array of images of the same scene [4]. A different way [3] is to use software and generate different views of the same imaged scene. This has the advantage of generating high quality images that can be used as ground truth in a variety of applications, such as depth or disparity estimation.

Another way of obtaining a 4-D light scene representation is by using a micro lens based plenoptic camera. The idea behind these cameras was developed by Ren Ng [56]. His idea was then put in practice when these revolutionary alternatives to conventional digital cameras became available to consumers (Lytro [59], Raytrix [60]). A comparison between how light in a scene is recorded on the camera sensor in plenoptic cameras and in traditional cameras is illustrated in Figure 4.1. Specifically, in the light field camera (shown in Figure 4.1b), an array of small lenses is placed between the camera lens and the camera sensor. These small lenses separate the converging rays and create separate images of the scene on the camera sensor. In this way, plenoptic cameras capture all light travelling along all directions intersecting the sensor [56], as opposed to their conventional counterparts (shown in Figure 4.1a), which by design capture the total amount of light at each point on the sensor.



(a) Conventional camera **(b)** Light field camera

Figure 4.1: Schematic camera representation

The advantage of this new way of representing light in a scene is the tremendous number of applications it opens the gate for. For photographers, the ability to change the focus after the photo was taken is often desired and this is possible when a plenoptic camera is used [56]. The ability to change the aperture of the camera is often useful in artistic/ visually pleasing scene representation. This in turn has interesting applications in reconstructing occluded surfaces [61]. Needless to say, eliminating occlusions from a scene has important benefits in security. Light field imaging has also potential in assisting devices such as robots that operate in notoriously difficult environments [62]. Designing high dynamic range video projectors is also possible if one makes use of light field information [63]. Another ongoing challenge that has been tackled in the light field domain is designing glasses-free 3-D displays [64] or displays that can compensate viewers' visual aberrations [65] can also be tackled if one is given a light field scene representation.

4.3.2 Signal of interest visualization

The light field notion as described earlier in this chapter has gained increased attention over the past two decades and as such the research community is still confronted with a series of open problems, such as understanding what is the most intuitive way for a user to visualize and manipulate such data. Recent user studies [2], [66] analyze two main interaction paradigms for visualizing and manipulating light fields: the multifocus and the multiview paradigm. In the first paradigm, the user sees and can distinguish between objects in the light field by synthetically changing the aperture of

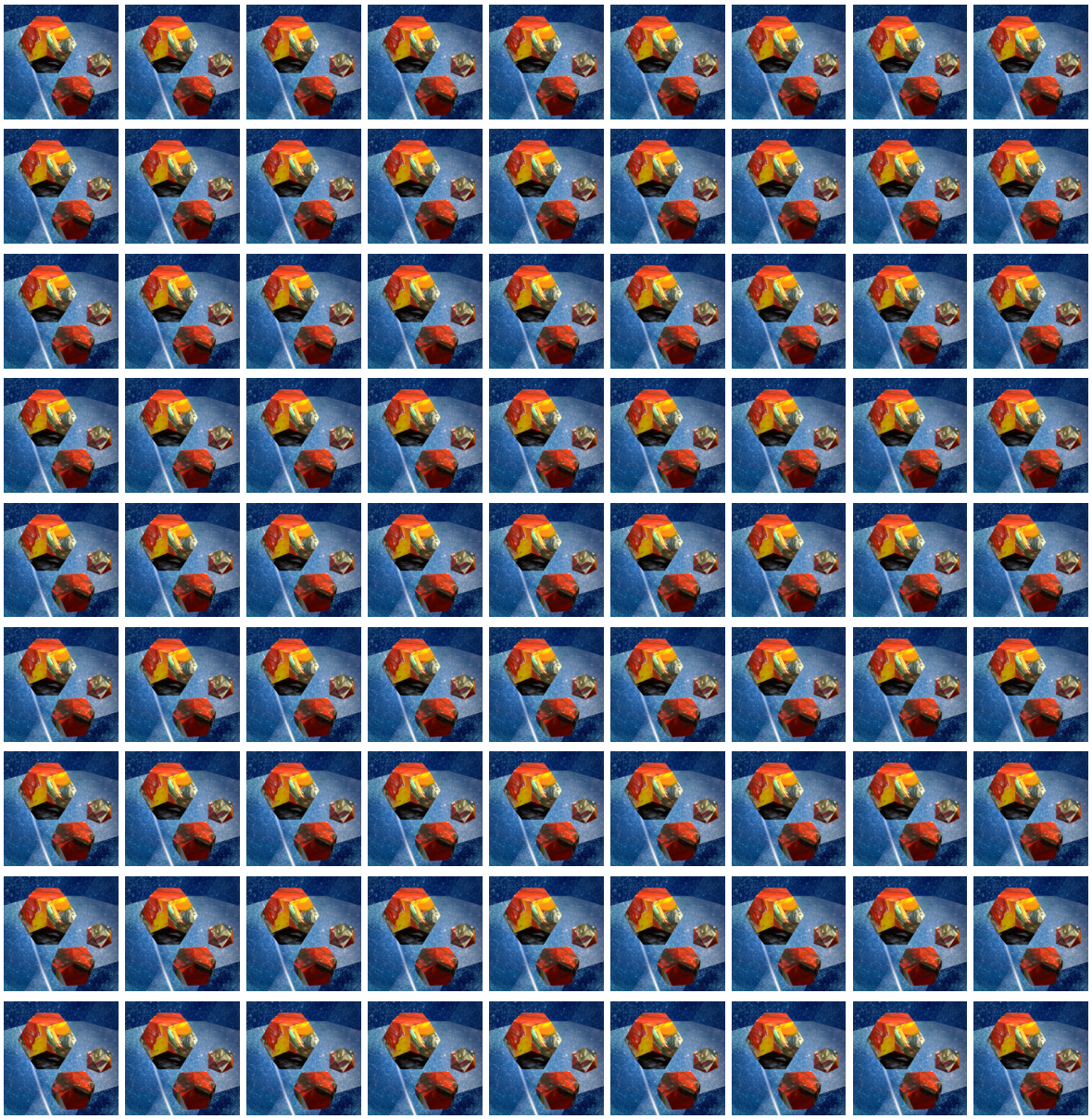


Figure 4.2: Multiview representation of (color) light field **platonic** [3]. The light field is pictured as a 9×9 array of $512 \times 512 \times 3$ images, with the last dimension indexing color information (RGB model considered)

the camera, generating a narrow depth of field with the object of interest in the centre. In the second paradigm, the user has access to the multiple views captured by the light field sensor, and this is the representation preferred in this work. Specifically, a light field is viewed as a 2-D array of (color or grayscale) images. Figure 4.2 illustrates the multiview representation of a computer generated light field [67] with size $9 \times 9 \times 512 \times 512 \times 3$. The first two signal dimensions correspond to the size of the array

(a) Image (1,1) (b) Image (1,5) (c) Image (1,9)

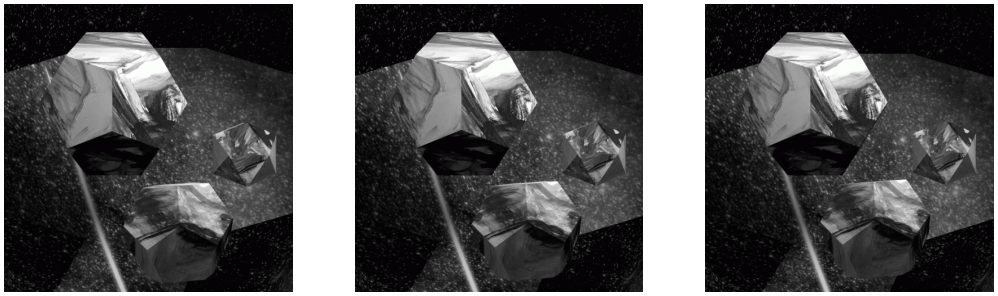
(d) Image (5,1) (e) Image (5,5) (f) Image (5,9)

(g) Image (9,1) (h) Image (9,5) (i) Image (9,9)

Figure 4.3: Sample images in a color light field. Original light field size: $9\times9\times512\times512\times3$

of images, and the following two correspond to the size of each individual image in the array. The last dimension indexes color information.

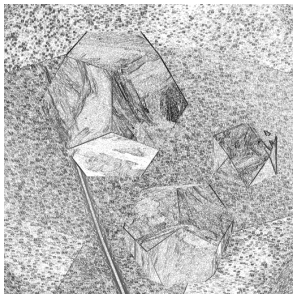
In Figure 4.3, a 3×3 array of images extracted from a benchmark light field **platonic** [3] is shown. Several images from the luminance channel of the light field are shown in Figure 4.4a–4.4c. As the light field is a function of four variables, its gradient has four directional components. Two of these components are shown in Figure 4.4d–4.4i. The multi-dimensional array Φ_x depicts horizontal changes in each image of the array, shown in Figure 4.4d–4.4f, while Φ_y depicts vertical changes, as seen in Figure 4.4g–4.4i. The multi-dimensional array Φ_u depicts horizontal changes between consecutive images in the array, shown in Figure 4.5a–4.5c, while Φ_v depicts vertical changes, as seen in Figure 4.5d–4.5f.



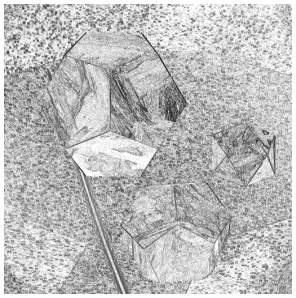
(a) Image (1,1)

(b) Image (5,5)

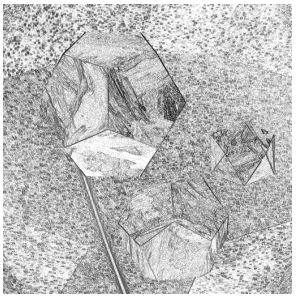
(c) Image (9,9)



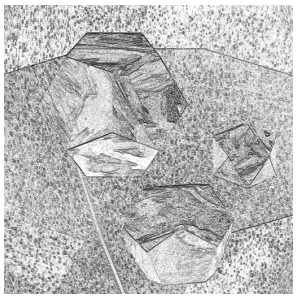
(d) Horizontal changes in image (1,1)



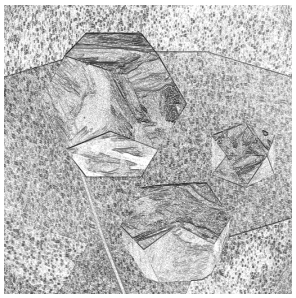
(e) Horizontal changes in image (5,5)



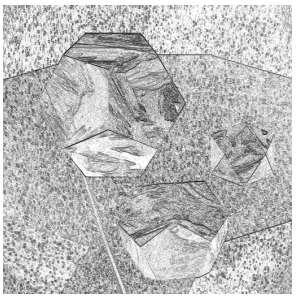
(f) Horizontal changes in image (9,9)



(g) Vertical changes in image (1,1)

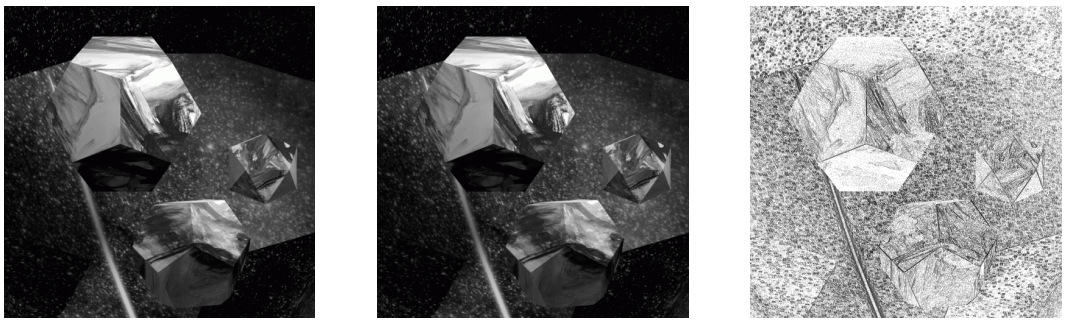


(h) Vertical changes in image (5,5)



(i) Vertical changes in image (9,9)

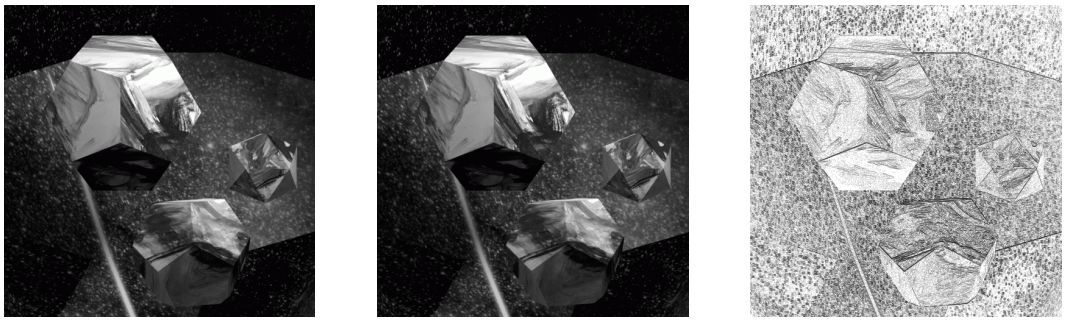
Figure 4.4: Visualization of the relationship between the images in the light field (top row), and the corresponding spatial derivatives Φ_x and Φ_y (middle, and bottom row, respectively). In the last two rows, black depicts a high magnitude derivative value.



(a) Image (1,1)

(b) Image (1,2)

(c) Change between image (1,2) and image (1,1)



(d) Image (1,1)

(e) Image (2,1)

(f) Change between image (2,1) and image (1,1)

Figure 4.5: Visualization of spatial derivatives Φ_u and Φ_v (i.e., the gradient components that depict changes between consecutive images in the array, in the horizontal and vertical direction, respectively). In the last column, black depicts a high magnitude derivative value.

4.4 Notation

4.4.1 Filtering

In what follows, $\Phi H(z)$ denotes one dimensional filtering of multi-dimensional signal Φ with a one dimensional filter given by transfer function $H(z)$. The subscript k in the expression $\Phi H(z_k)$ indicates that the filtering is performed in the direction d_k.

For the purpose of making this chapter self-contained, the definition of Haar wavelet analysis filters is recalled here:

$$H_L(z_i) = \frac{1}{\sqrt{2}}\left(1 + z_i^{-1}\right) \tag{4.1}$$

$$H_H(z_i) = \frac{1}{\sqrt{2}}\left(1 - z_i^{-1}\right) \tag{4.2}$$

where $i \in \{1, 2, 3, 4\}$ denotes the filtering direction.

4.4.2 Sampling rate change

Directional downsampling by a factor n along direction d_k is denoted by $\downarrow_{n,d_k}$. For example, downsampling a 4-D signal Φ by 2 along the first direction is denoted $\downarrow_{2,d_1} \Phi$. Directional upsampling by a factor n along direction d_k is denoted by $\uparrow_{n,d_k}$.

Downsampling or upsampling along more than one direction is indicated by placing the corresponding directional subscripts one after the other. For example, downsampling signal Φ by 2 along directions d_1 and d_3 is denoted $\downarrow_{2,d_1d_3} \Phi$.

When downsampling or upsampling is done along all directions, the directional subscripts are dropped, to simplify the notation. For example, downsampling signal Φ by a factor of 2 along all four directions is denoted by $\downarrow_2 \Phi$.

4.5 Four dimensional signal reconstruction from gradient

The problem at hand here is to produce a 4-D signal $\Phi \in \mathbb{R}^{2^{d_1} \times 2^{d_2} \times 2^{d_3} \times 2^{d_4}}$ from a given gradient data set and an estimate of the average value of the signal. Without the loss of generality, the algorithm is developed under the assumption that $0 < d_1 \leqslant d_2 \leqslant d_3 \leqslant d_4$.

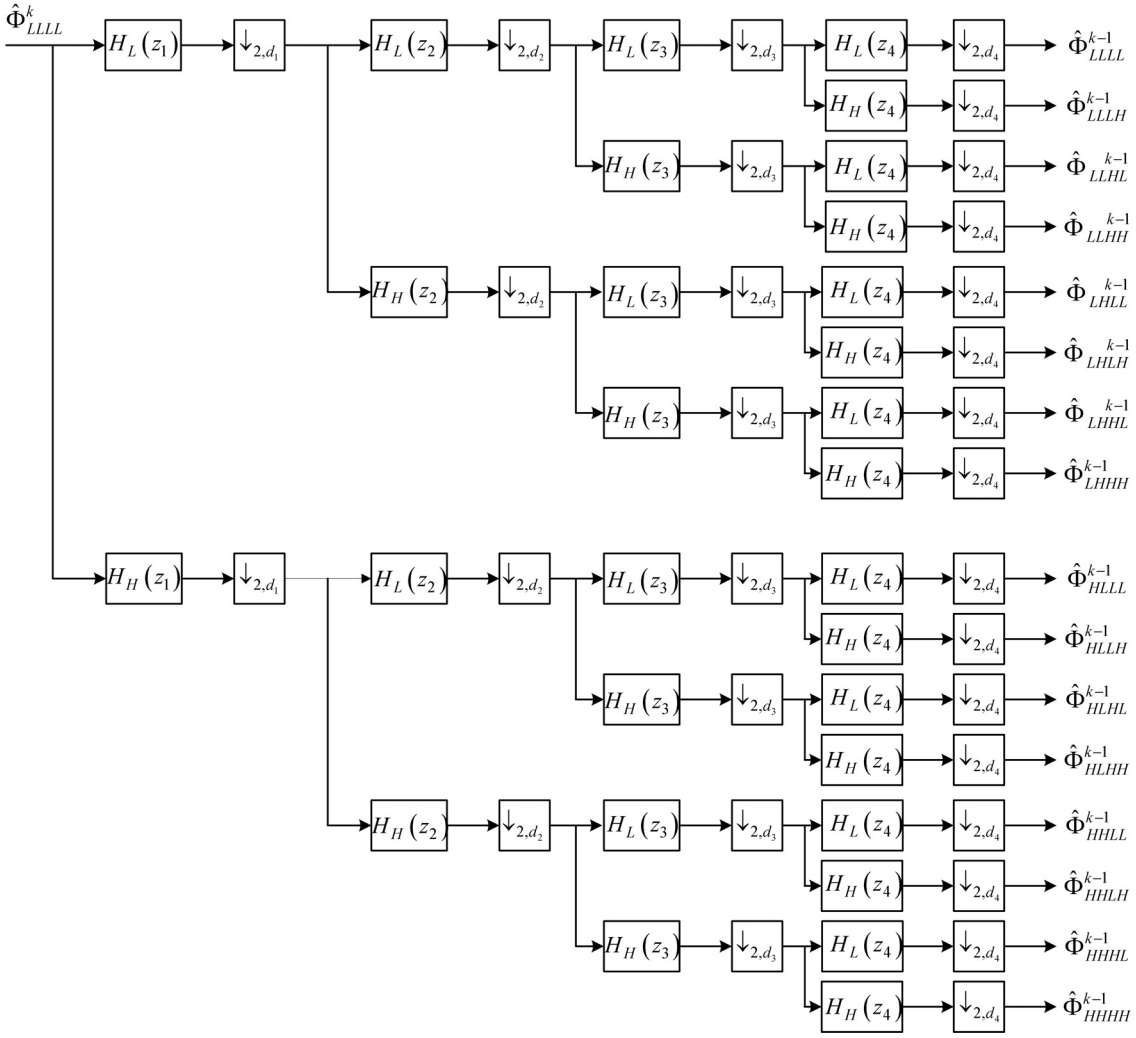


Figure 4.6: One step in the (Haar) wavelet decomposition of a 4-D signal

The algorithm is developed in a similar framework with the 2-D and 3-D algorithms presented in 2 and 3, and has two main steps. First, an analysis step, at the end of which the 4-D Haar wavelet decomposition is obtained from the given gradient data. Second, a synthesis step, in which a signal is obtained from the Haar wavelet decomposition.

The (full) Haar wavelet decomposition of a signal with size $2^{d_1} \times 2^{d_2} \times 2^{d_3} \times 2^{d_4}$ is obtained from the signal in d_1 steps. One step of this process is illustrated in Figure 4.6. Concretely, the signal is downsampled and filtered as shown, generating the 16 subsignals (subband coefficients) illustrated on the right hand side of Figure 4.6. This process is then repeated to signal $\hat{\Phi}^{k-1}_{LLLL}$. The full decomposition is obtained after d_1 steps, with the convention that $\hat{\Phi}^{d_1}_{LLLL}$ is the full resolution signal, and $\hat{\Phi}^{0}_{LLLL}$

is the lowest resolution.

In contrast to finding the Haar wavelet decomposition from the signal, presented here is a means to obtain the Haar wavelet decomposition from the gradient of the signal. Similarly to the process illustrated in Figure 4.6, the process of finding the wavelet decomposition from the gradient data is iterative, and will be described in the following subsections.

4.5.1 Analysis step: detail subbands of the wavelet decomposition

Finding the wavelet decomposition of the signal of interest from the derivatives begins by finding 2^4-1 detail subbands: $\hat{\Phi}^{d_1-1}_{HHHH}$, $\hat{\Phi}^{d_1-1}_{HHHL}$, $\hat{\Phi}^{d_1-1}_{HHLH}$, $\hat{\Phi}^{d_1-1}_{HLHH}$, $\hat{\Phi}^{d_1-1}_{LHHH}$, $\hat{\Phi}^{d_1-1}_{HHLL}$, $\hat{\Phi}^{d_1-1}_{HLHL}$, $\hat{\Phi}^{d_1-1}_{HLLH}$, $\hat{\Phi}^{d_1-1}_{LHHL}$, $\hat{\Phi}^{d_1-1}_{LHLH}$, $\hat{\Phi}^{d_1-1}_{LLHH}$, $\hat{\Phi}^{d_1-1}_{HLLL}$, $\hat{\Phi}^{d_1-1}_{LHLL}$, $\hat{\Phi}^{d_1-1}_{LLHL}$, and $\hat{\Phi}^{d_1-1}_{LLLH}$.

For this purpose, it is useful to recall the connection between the gradient components of a 4-D signal Φ and the 1-D Haar highpass filter $H_H(z)$:

$$\Phi_i = \sqrt{2}\Phi H_H(z_i) \tag{4.3}$$

Φ_i are the four directional components of the gradient of Φ, and $i \in \{1,2,3,4\}$. Recall also that $\Phi H_H(z_i)$ indicates filtering Φ along direction d_i.

At resolution d_1-1, the detail coefficients can be found by evaluating the following expressions:

$$\hat{\Phi}^{d_1-1}_{HHHH} = \downarrow_{2,d_3}\left\{\left[\downarrow_{2,d_4}\left\{\left[\downarrow_{2,d_1}\left\{\left[\downarrow_{2,d_2}\left\{\frac{1}{\sqrt{2}}\Phi_2\right\}\right]H_H(z_1)\right\}\right]H_H(z_4)\right\}\right]H_H(z_3)\right\} \tag{4.4}$$

$$\hat{\Phi}^{d_1-1}_{HHHL} = \downarrow_{2,d_3}\left\{\left[\downarrow_{2,d_4}\left\{\left[\downarrow_{2,d_1}\left\{\left[\downarrow_{2,d_2}\left\{-\frac{1}{\sqrt{2}}\Phi_2\right\}\right]H_H(z_1)\right\}\right]H_H(z_4)\right\}\right]H_L(z_3)\right\} \tag{4.5}$$

$$\hat{\Phi}^{d_1-1}_{HHLH} = \downarrow_{2,d_3}\left\{\left[\downarrow_{2,d_4}\left\{\left[\downarrow_{2,d_1}\left\{\left[\downarrow_{2,d_2}\left\{-\frac{1}{\sqrt{2}}\Phi_2\right\}\right]H_H(z_1)\right\}\right]H_L(z_4)\right\}\right]H_H(z_3)\right\} \tag{4.6}$$

$$\hat{\Phi}^{d_1-1}_{HHLL} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_1} \left\{ \left[\downarrow_{2,d_2} \left\{ \frac{1}{\sqrt{2}} \Phi_{d_2} \right\} \right] H_H(z_1) \right\} \right] H_L(z_4) \right\} \right] H_L(z_3) \right\} \tag{4.7}$$

$$\hat{\Phi}^{d_1-1}_{HLHH} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_1} \left\{ \left[\downarrow_{2,d_2} \left\{ -\frac{1}{\sqrt{2}} \Phi_2 \right\} \right] H_L(z_1) \right\} \right] H_H(z_4) \right\} \right] H_H(z_3) \right\} \tag{4.8}$$

$$\hat{\Phi}^{d_1-1}_{HLHL} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_1} \left\{ \left[\downarrow_{2,d_2} \left\{ \frac{1}{\sqrt{2}} \Phi_2 \right\} \right] H_L(z_1) \right\} \right] H_H(z_4) \right\} \right] H_L(z_3) \right\} \tag{4.9}$$

$$\hat{\Phi}^{d_1-1}_{HLLH} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_1} \left\{ \left[\downarrow_{2,d_2} \left\{ \frac{1}{\sqrt{2}} \Phi_2 \right\} \right] H_L(z_1) \right\} \right] H_L(z_4) \right\} \right] H_H(z_3) \right\} \tag{4.10}$$

$$\hat{\Phi}^{d_1-1}_{HLLL} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_1} \left\{ \left[\downarrow_{2,d_2} \left\{ -\frac{1}{\sqrt{2}} \Phi_2 \right\} \right] H_L(z_1) \right\} \right] H_L(z_4) \right\} \right] H_L(z_3) \right\} \tag{4.11}$$

$$\hat{\Phi}^{d_1-1}_{LHHH} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_2} \left\{ \left[\downarrow_{2,d_1} \left\{ -\frac{1}{\sqrt{2}} \Phi_1 \right\} \right] H_L(z_2) \right\} \right] H_H(z_4) \right\} \right] H_H(z_3) \right\} \tag{4.12}$$

$$\hat{\Phi}^{d_1-1}_{LHHL} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_2} \left\{ \left[\downarrow_{2,d_1} \left\{ \frac{1}{\sqrt{2}} \Phi_1 \right\} \right] H_L(z_2) \right\} \right] H_H(z_4) \right\} \right] H_L(z_3) \right\} \tag{4.13}$$

$$\hat{\Phi}^{d_1-1}_{LHLH} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_2} \left\{ \left[\downarrow_{2,d_1} \left\{ \frac{1}{\sqrt{2}} \Phi_1 \right\} \right] H_L(z_2) \right\} \right] H_L(z_4) \right\} \right] H_H(z_3) \right\} \tag{4.14}$$

$$\hat{\Phi}_{LHLL}^{d_1-1} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_2} \left\{ \left[\downarrow_{2,d_1} \left\{ -\frac{1}{\sqrt{2}} \Phi_1 \right\} \right] H_L(z_2) \right\} \right] H_L(z_4) \right\} \right] H_L(z_3) \right\} \tag{4.15}$$

$$\hat{\Phi}_{LLHH}^{d_1-1} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_1} \left\{ \left[\downarrow_{2,d_2} \left\{ \left[\downarrow_{2,d_4} \left\{ \frac{1}{\sqrt{2}} \Phi_4 \right\} \right] H_L(z_2) \right\} \right] H_L(z_1) \right\} \right] H_H(z_3) \right\} \tag{4.16}$$

$$\hat{\Phi}_{LLHL}^{d_1-1} = \downarrow_{2,d_3} \left\{ \left[\downarrow_{2,d_1} \left\{ \left[\downarrow_{2,d_2} \left\{ \left[\downarrow_{2,d_4} \left\{ -\frac{1}{\sqrt{2}} \Phi_4 \right\} \right] H_L(z_2) \right\} \right] H_L(z_1) \right\} \right] H_L(z_3) \right\} \tag{4.17}$$

$$\hat{\Phi}_{LLLH}^{d_1-1} = \downarrow_{2,d_4} \left\{ \left[\downarrow_{2,d_1} \left\{ \left[\downarrow_{2,d_2} \left\{ \left[\downarrow_{2,d_3} \left\{ -\frac{1}{\sqrt{2}} \Phi_3 \right\} \right] H_L(z_2) \right\} \right] H_L(z_1) \right\} \right] H_L(z_4) \right\} \tag{4.18}$$

To completely determine the first level of the wavelet decomposition, the remaining subband $\hat{\Phi}_{LLLL}^{d_1-1}$ needs to be found. The approach described here for this task is recursive, and entails finding the 15 detail subbands at consecutive resolution levels, starting from level $d_1 - 2$ down to 0. All **detail** subbands can be found from the gradient data, and a way to do this is described below.

The following notation helps to achieve a more compact form of the equations:

$$\Phi_i^F = \frac{1}{\left(\sqrt{2}\right)^3} \Phi_i \prod_{m \neq i} H_L(z_m) \tag{4.19}$$

where $i, m \in \{1, 2, 3, 4\}$ and Φ_i is a directional component of the given gradient data. Then, for $k = 2$ to d_1, the 15 subbands are given by:

$$\hat{\Phi}_{HHHH}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_1^F H_L^2(z_1) H_H\left(z_2^2\right) H_H\left(z_3^2\right) H_H\left(z_4^2\right) \right\} \tag{4.20}$$

$$\hat{\Phi}_{HHHL}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_1^F H_L^2(z_1) H_H\left(z_2^2\right) H_H\left(z_3^2\right) H_L\left(z_4^2\right) \right\} \tag{4.21}$$

$$\hat{\Phi}_{HHLH}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_1^F H_L^2(z_1) H_H\left(z_2^2\right) H_L\left(z_3^2\right) H_H\left(z_4^2\right) \right\} \tag{4.22}$$

$$\hat{\Phi}_{HLHH}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_1^F H_L^2\left(z_1\right) H_L\left(z_2^2\right) H_H\left(z_3^2\right) H_H\left(z_4^2\right) \right\} \tag{4.23}$$

$$\hat{\Phi}_{LHHH}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_2^F H_L\left(z_1^2\right) H_L^2\left(z_2\right) H_H\left(z_3^2\right) H_H\left(z_4^2\right) \right\} \tag{4.24}$$

$$\hat{\Phi}_{HHLL}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_1^F H_L^2\left(z_1\right) H_H\left(z_2^2\right) H_L\left(z_3^2\right) H_L\left(z_4^2\right) \right\} \tag{4.25}$$

$$\hat{\Phi}_{HLHL}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_1^F H_L^2\left(z_1\right) H_L\left(z_2^2\right) H_H\left(z_3^2\right) H_L\left(z_4^2\right) \right\} \tag{4.26}$$

$$\hat{\Phi}_{HLLH}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_1^F H_L^2\left(z_1\right) H_L\left(z_2^2\right) H_L\left(z_3^2\right) H_H\left(z_4^2\right) \right\} \tag{4.27}$$

$$\hat{\Phi}_{LHHL}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_2^F H_L\left(z_1^2\right) H_L^2\left(z_2\right) H_H\left(z_3^2\right) H_L\left(z_4^2\right) \right\} \tag{4.28}$$

$$\hat{\Phi}_{LHLH}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_2^F H_L\left(z_1^2\right) H_L^2\left(z_2\right) H_L\left(z_3^2\right) H_H\left(z_4^2\right) \right\} \tag{4.29}$$

$$\hat{\Phi}_{LLHH}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_3^F H_L\left(z_1^2\right) H_L\left(z_2^2\right) H_L^2\left(z_3\right) H_H\left(z_4^2\right) \right\} \tag{4.30}$$

$$\hat{\Phi}_{HLLL}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_1^F H_L^2\left(z_1\right) H_L\left(z_2^2\right) H_L\left(z_3^2\right) H_L\left(z_4^2\right) \right\} \tag{4.31}$$

$$\hat{\Phi}_{LHLL}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_2^F H_L\left(z_1^2\right) H_L^2\left(z_2\right) H_L\left(z_3^2\right) H_L\left(z_4^2\right) \right\} \tag{4.32}$$

$$\hat{\Phi}_{LLHL}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_3^F H_L\left(z_1^2\right) H_L\left(z_2^2\right) H_L^2\left(z_3\right) H_L\left(z_4^2\right) \right\} \tag{4.33}$$

$$\hat{\Phi}_{LLLH}^{d_1-k} = \downarrow_4 \left\{ \left(\sqrt{2}\right)^3 \Phi_4^F H_L\left(z_1^2\right) H_L\left(z_2^2\right) H_L\left(z_3^2\right) H_L^2\left(z_4\right) \right\} \tag{4.34}$$

Update the coefficients Φ_i^F

$$\Phi_i^F = \downarrow_2 \left\{ \sqrt{2} \Phi_i^F H_L\left(z_i\right) \prod_{m \neq i} H_L\left(z_m^2\right) \right\}$$

where $i, m \in \{1, 2, 3, 4\}$

Once the process described by Equations 4.20–4.34 is complete, all the detail subbands in the wavelet decomposition of Φ have been found. What is missing to complete the wavelet decomposition is the lowest resolution subband, i.e. $\hat{\Phi}^0_{LLLL}$. A means to recover these coefficients is described in the following subsection.

4.5.2 Analysis step: approximation subband at the lowest resolution

4.5.2.1 Signal with equal dimensions

When the signal dimensions are all equal, i.e. when the signal has size $2^{d_1} \times 2^{d_1} \times 2^{d_1} \times 2^{d_1}$, the approximation subband at the lowest resolution is a constant, proportional to the mean value of the signal. Specifically:

$$\hat{\Phi}^0_{LLLL} = \left(\sqrt{2}\right)^{4d_1} \cdot m \tag{4.35}$$

where m is the average value of the signal.

In other words, while it is not possible to find the lowest subband from the signal derivatives, it can be approximated from the average value of the signal.

4.5.2.2 Signal with different dimensions

The question this section will provide an answer to is how to find the lowest resolution subband *when the signal dimensions are not equal*.

The analysis and findings presented in the remainder of this chapter assume that the unknown 4-D signal has a light field structure. In particular, the similarity between all images in the array is the key property exploited in the developed method.

First, the derivatives corresponding to the central image in the light field are extracted from the light field derivative data set. Then, an approximation of the central image in the light field is obtained from the extracted derivative, using the 2-D integration algorithm described in Chapter 2. An approximation of $\hat{\Phi}^0_{LLLL}$ (of the entire light field) can then be found by summing all the elements in square consecutive regions of the central image, with the side of the square equal to the smallest dimension of the light field, e.g. 4 in a $4 \times 4 \times 64 \times 128$ light field.

The technique described above gives only an approximation of $\hat{\Phi}^0_{LLLL}$, so after the analysis step is complete, the decomposition obtained will be an approximation of the

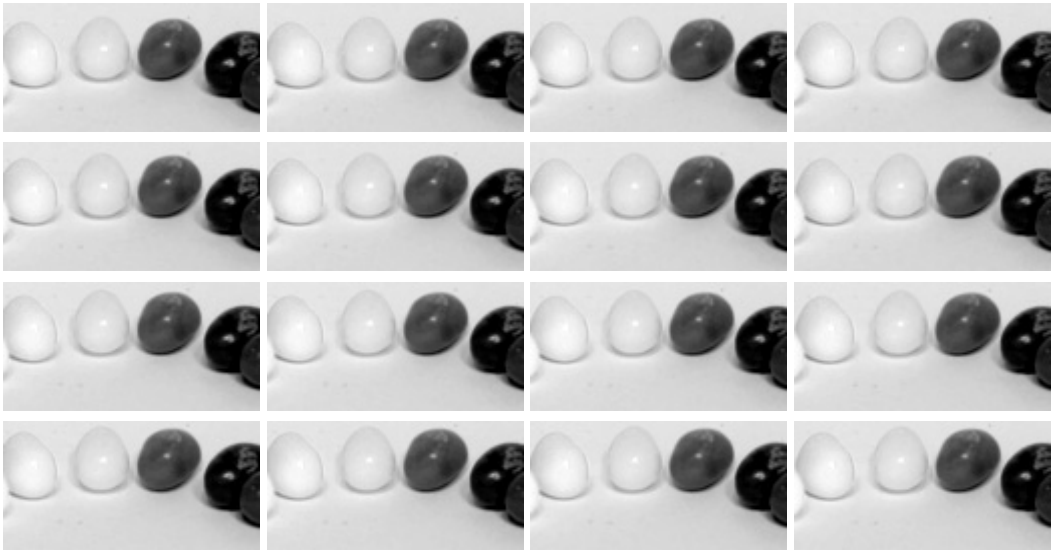


Figure 4.7: A $4 \times 4 \times 64 \times 128$ slice from the light field **jellybeans** [4]

"true" wavelet decomposition of the signal. Therefore, an error will be introduced in the reconstructed signal. This error is smoothened out during the synthesis step, when a couple of iterations of the Poisson solver are used at each resolution level.

An illustrative example

A light field with size $4 \times 4 \times 64 \times 128$ was generated from benchmark light field **jellybeans** [4]. The light field is shown in Figure 4.7 as a 4×4 matrix of images, each having size 64×128.

A first impression on how good the approximation method is can be made by studying Figure 4.8. On the top row, a comparison between the lowest resolution subband of the wavelet decomposition of the light field shown in 4.7 and the approximation obtained from the central image as described is shown. Note that the two sets of subband coefficients are quite similar. The bottom left hand side of Figure 4.8 shows the absolute value of the difference between the two sets of subband coefficients, normalized with respect to the absolute value of the "true" $\hat{\Phi}^0_{LLLL}$. For completeness, the bottom right plot shows the central image in the light field.

Depending on the signal size and content, the approximation technique can be attempted in one of the patterns illustrated in Figure 4.9. The impact of approximating the lowest resolution subband $\hat{\Phi}^0_{LLLL}$ on the quality of the reconstructed light fields will be studied in 4.6.

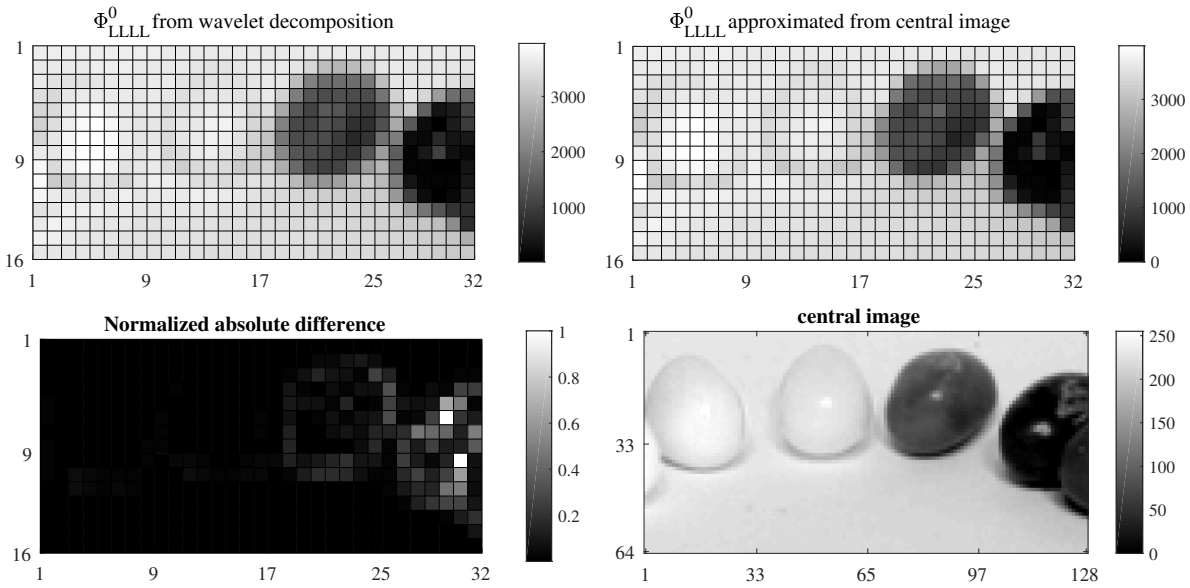
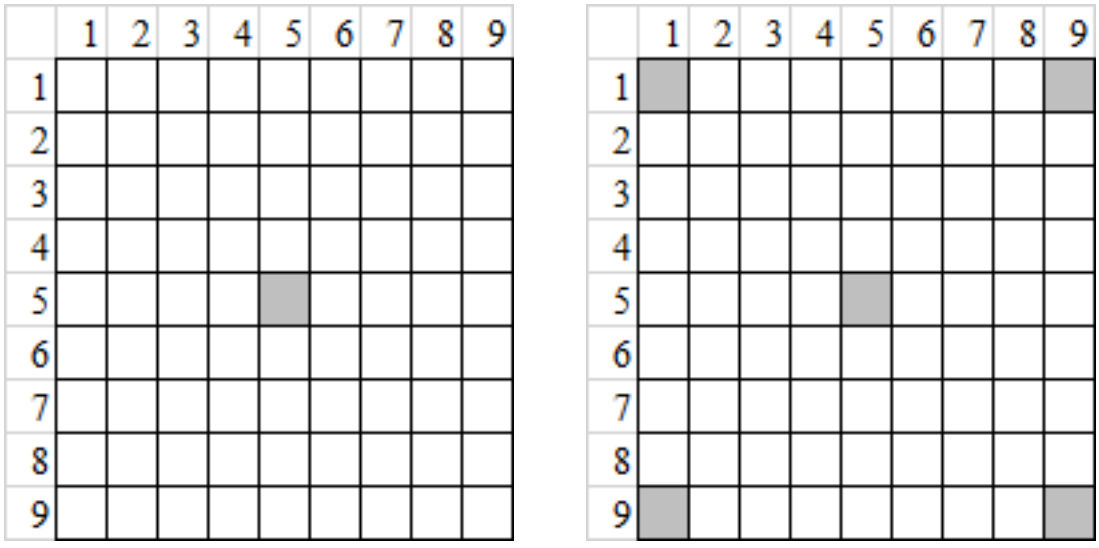


Figure 4.8: Visualizing the $\hat{\Phi}^0_{LLLL}$ subband of the wavelet decomposition of a light field Φ with size $4 \times 4 \times 64 \times 128$. Size of $\hat{\Phi}^0_{LLLL}$ is $1 \times 1 \times 16 \times 32$ and is shown on the top row as a 2-D signal. The bottom left image depicts the absolute value of the difference between the "true" $\hat{\Phi}^0_{LLLL}$, obtained from the wavelet decomposition, and its approximation obtained from the central image (normalized with respect to the absolute value of the "true" $\hat{\Phi}^0_{LLLL}$). The bottom right plot shows the central image in the light field.



(a) Pattern "One image" **(b)** Pattern "Five images"

Figure 4.9: Illustration of images used to approximate lowest resolution subband. The light field is shown as a 9×9 array of images and the highlighted squares are the images used to approximate $\hat{\Phi}^0_{LLLL}$

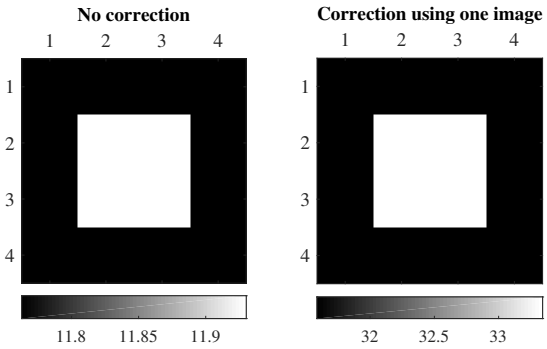


Figure 4.10: Matrix of PSNR values (dB) between images in the original and in the reconstructed light field. A lower value indicates better quality of the image in the reconstructed light field. A higher value indicates lower quality of the image in the reconstructed light field.

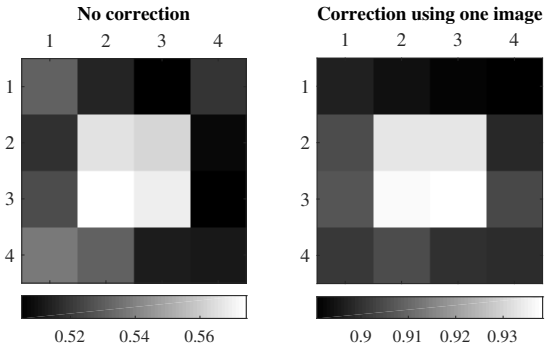


Figure 4.11: Structural similarity index measure [17] between original images and the ones in the reconstructed light field. A low value (blue) indicates low quality of the image in the reconstructed light field. A high value indicates high quality of the image in the reconstructed light field. The index range is from 0 to 1, with 1 corresponding to the image most similar to the given one

To assess the effectiveness of the technique presented, the light field was reconstructed from its derivatives in two different ways: first, the 16 entries of $\hat{\Phi}^0_{LLLL}$, were considered equal; second, an approximation of the 16 entries was obtained from image $(3, 3)$ in the (4,4) array. Figures 4.10 and 4.11 show the PSNR and structural similarity index measure (SSIM) [17] between the images in the original light field and the images in the reconstruction.

4.5.3 Synthesis step

Once the signal wavelet decomposition is obtained from the signal gradient as described in Sections 4.5.1–4.5.2, the 4-D signal is reconstructed using a slightly modified version of the 4-D Haar wavelet synthesis process. The modification introduced in the (standard) wavelet synthesis is necessary in order to visually correct, or smoothen out, the errors introduced in the reconstruction by approximating the final subband in the wavelet decomposition of the light field of interest. This correction is done by using a 4-D extension of the iterative Poisson solver described in Appendix B at each resolution.

4.5.4 Handling "non-power of two" dimensions

The most attractive feature of the signal reconstruction technique presented in this chapter is that it can be used to to obtain a (4-D) signal from a gradient data set, without the need to extend the given gradient data set to that corresponding to a hypercube (i.e. a 4-D signal with all dimensions equal). As described earlier, the first step of the proposed reconstruction technique is to generate the wavelet decomposition of the signal of interest. As such, there is an implicit requirement on each of the dimensions of the signal, and that requirement is that they have to be a power of two. As such, a brief discussion about how the non-power of two case can be handled is in order, and is presented here.

A possible approach is to extend the gradient data set to a set corresponding to a hypercube with all dimensions equal to the largest of the nearest power of two of the given data. That approach has non-negligible consequences on the size of the data and is not taken here. Instead, a means was devised to extend the given gradient to that corresponding to a signal in which all dimensions are powers of two. For example, the gradient of a signal with size $5 \times 5 \times 250 \times 484$ is extended to that of a signal with size $8 \times 8 \times 256 \times 512$. The wavelet decomposition is then found from the extended version of the given gradient data set. An $8 \times 8 \times 256 \times 512$ signal is then obtained by modified wavelet synthesis. The signal of interest is then extracted from the extended signal.

There are two important aspects to be mentioned about the way in which this extension is done. The first one is that the extension step must not introduce border artifacts in the reconstructed signal. This is carried out by ensuring that the extended gradient data set satisfies the zero curl condition in the regions where the

extension was performed. The second aspect worth mentioning concerns the location of the original gradient dataset in the extended gradient dataset. Although slightly more cumbersome in terms of software implementation, placing the original gradient dataset in the centre of the extended version tends to produce better quality results, and this was the approach in the results reported here.

4.6 Performance evaluation

4.6.1 Indices

Although of a higher dimensional nature, the signals of interest in this chapter come from the realm of image processing. Finding a measure that ranks the quality of digital images in the same way as a human assessor would is a topic of research in itself and numerous indices were proposed and studied. To date there is no one index that suits all image processing and computer vision applications. The consensus in literature, however, when it comes to visual quality assessment seems to lean towards peak signal to noise ratio (PSNR) and structural similarity index measure (SSIM) [17], when a reference image is available. The approach here to assess the ability of the devised algorithm to reconstruct 4-D light fields from a gradient dataset is to compare corresponding views from the reconstructed and original light field, in terms of PSNR and SSIM.

For example, the quality of a reconstructed light field with size $5 \times 5 \times 256 \times 512$ is determined by comparing each of the 25 views from the reconstructed light field with each corresponding view from the original light field. For each of the 25 image pairs the PSNR and SSIM values are computed and a 5×5 matrix is generated for each index. The average value of each of these two matrices is then used as an overall indicator of the reconstructed light field quality. In addition, zoomed in regions of representative images in the light fields are studied to assist with the visual quality assessment.

4.6.2 Results

4.6.2.1 Results on 5×5 arrays of images

All light fields in the Synthetic Light Field Archive dataset [1] are computer generated, with two main purposes: to simulate light fields inside a camera (light fields **dragons**

and bunnies, **shrubbery**, **Messerschmitt**, **dice** and **fish**) and to develop new technologies such as glasses-free 3-D displays [63] (light fields **bunnies**, **TRex** and **Teapot**). The similarity between the images in the display light fields is usually lower than the camera light field displays, as they span a wider field of view with the intention of simulating a viewing zone in front of a glasses-free 3-D display.

On this dataset, in terms of PSNR, the approximation obtained using one image outperforms the one using five images on all but three light fields: **shrubbery**, **T-rex** and **teapot**. In terms of SSIM, the approximation obtained using one image outperforms the one using five images on all but three light fields: **dice**, **fish** and **teapot**.

In the **dragons and bunnies** light field, the artifacts are minimal, and visually perceived as variations in shading. The top row of Figure 4.15 shows a cropped 300×300 region from image (3,5) in the light field.

A similar observation can be made for the **shrubbery** light field. Some shading artifacts appear a bit more noticeable in the 300×300 region from image (3,5). Of note is the fact that the approximation using five images gives better quality results than the approximation using only the central image, and this is also confirmed by higher PSNR and SSIM values.

Both approximation techniques perform well on the light field **Messerschmitt**, with minimal visible artifacts.

In the **dice** light field, artifacts are visible as blocks around the main edges in the image, as can be seen on the fourth row of Figure 4.15. Note that this image contains little to no high frequencies (no fine details), and has mainly flat areas of intensity.

Table 4.1: Light fields with a 5×5 array structure from [1]

Light field ID	Light field name	Individual image size
1	Dragons and bunnies (aperture 3)	593×840
2	Shrubbery (aperture 20)	593×840
3	Messerschmitt (aperture 20)	593×840
4	Dice (aperture 20)	593×840
5	Fish (aperture 20)	593×840
6	Dragons and bunnies	525×840
7	T-Rex	525×840
8	Teapot	525×840

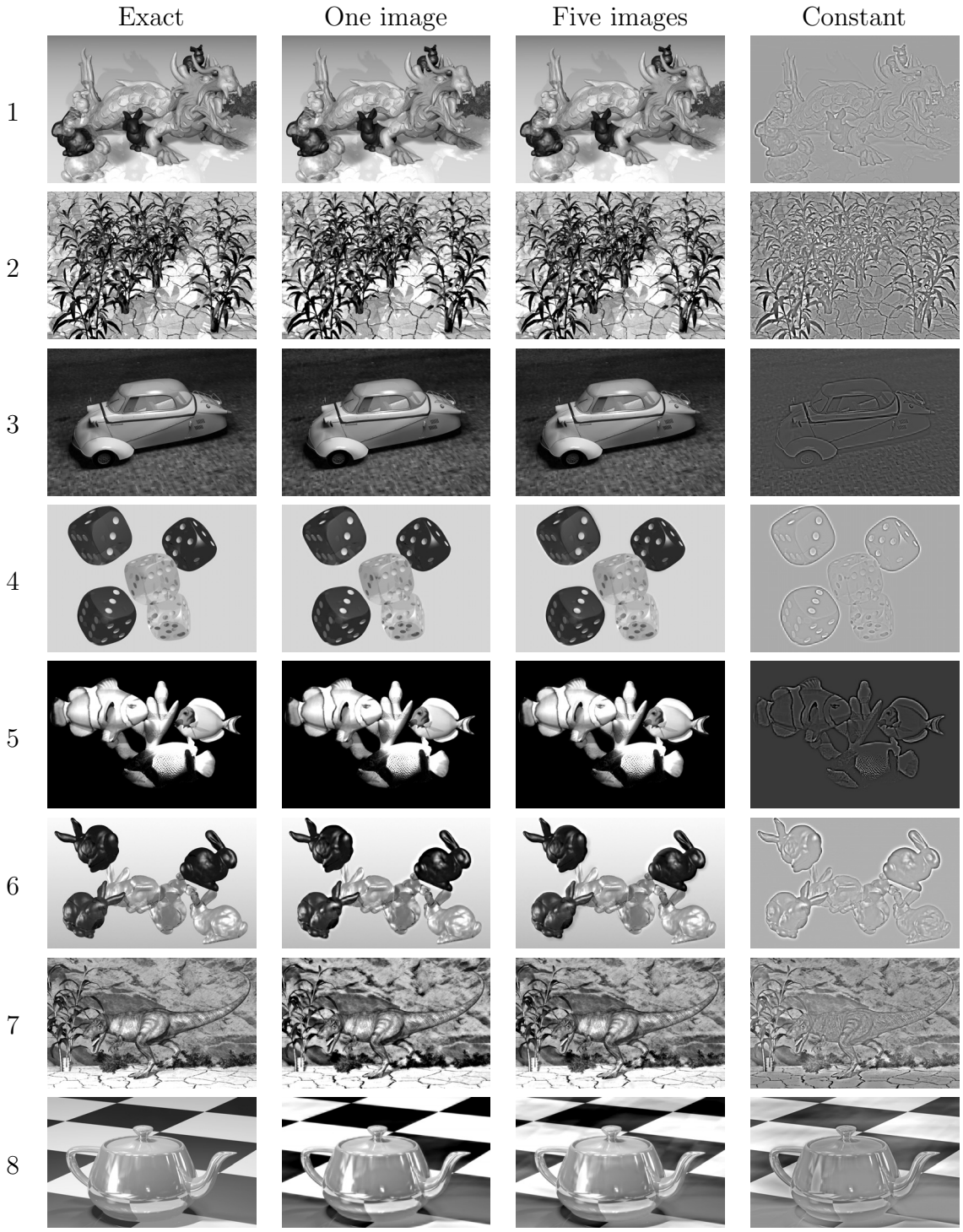
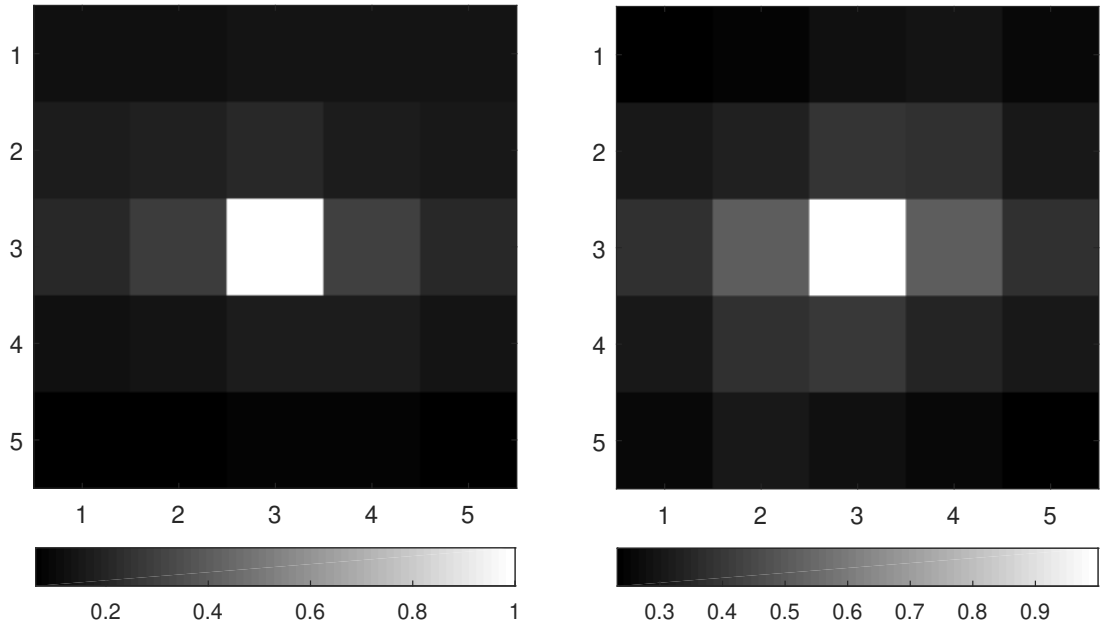


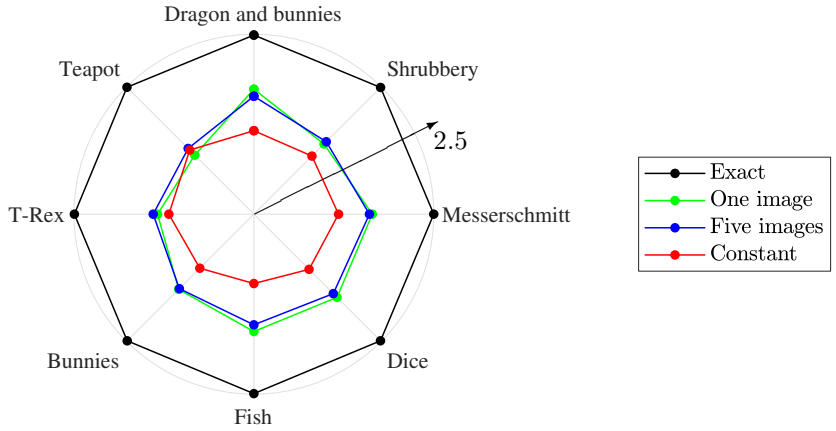
Figure 4.12: Central views of light fields listed in Table 4.1. Results from left to right on each row show the central views of the following light fields: reconstructed from the derivatives, and taking $\hat{\Phi}^0_{LLLL}$ from the wavelet decomposition of the light field; reconstructed from the derivatives, using the gradient of only one image in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, using the gradient of five images in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, approximating all entries of $\hat{\Phi}^0_{LLLL}$ with a constant.



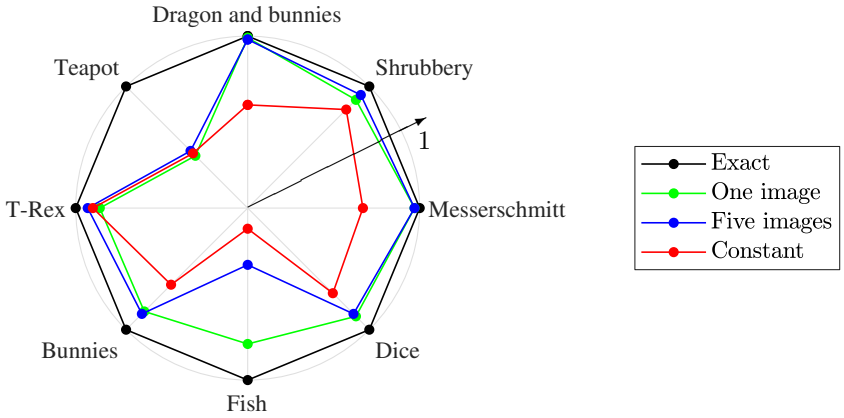
(a) SSIM map of **teapot**.

(b) SSIM map of **Messerschmitt**.

Figure 4.13: Similarity maps of images in light fields **teapot** and **Messerschmitt**. The central image was regarded as reference and the SSIM of all other images in the light field was computed with respect to it.



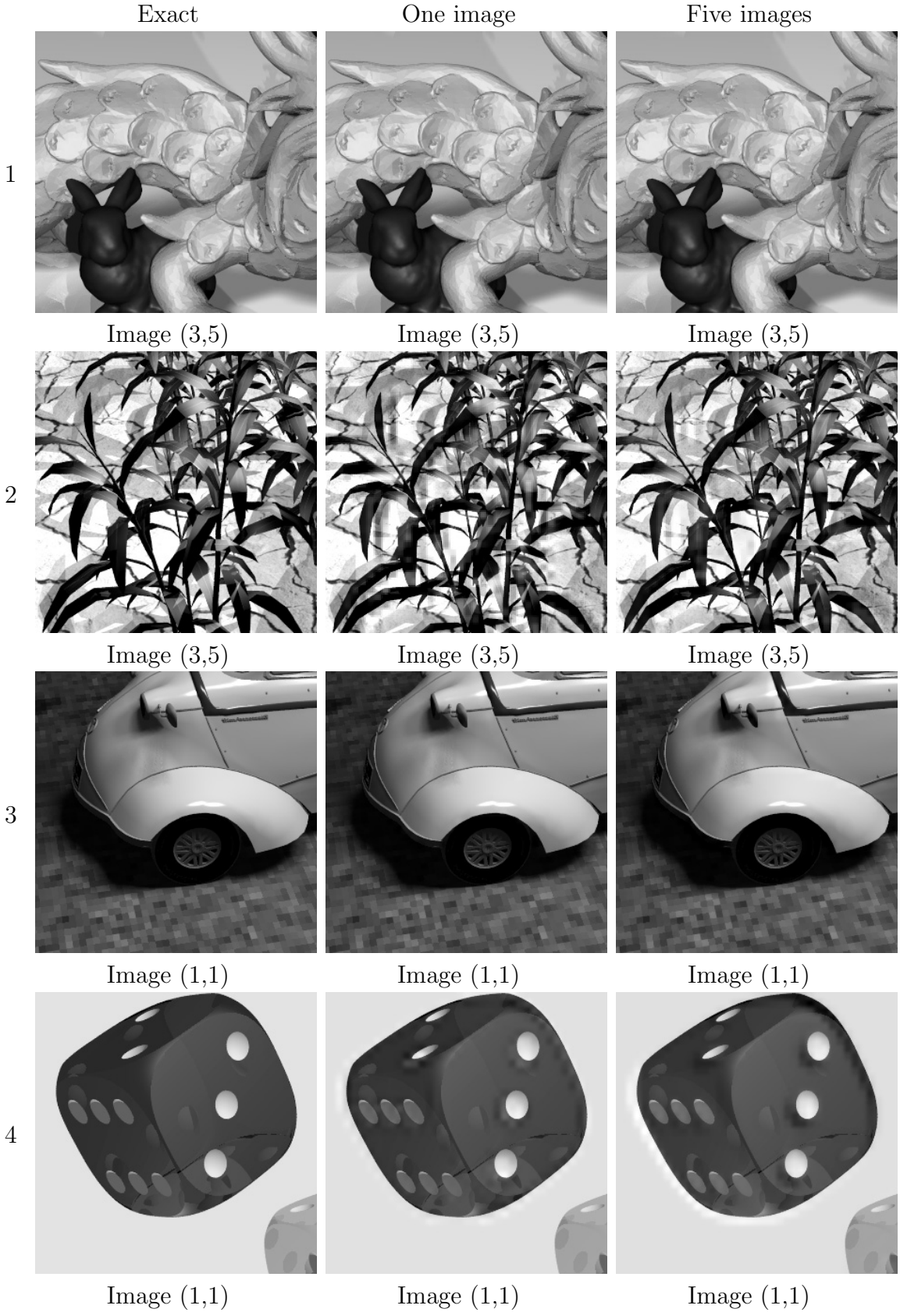
(a) Logarithmic representation of average PSNR values of reconstructed versus original light fields. The values farthest from the center denote a better quality of the reconstructed images.



(b) Linear representation of average SSIM values between reconstructed and original light fields. Values closest to 1 indicate better quality of the reconstructed images.

Figure 4.14: Algorithm performance on 5×5 arrays from database [1].PSNR and SSIM values of all images in the light fields listed in Table 4.1, reconstructed from their gradient, compared against their original counterparts from database [1].

In the **fish** light field, some artifacts are visible as blocks around the sharp image transitions, as can be seen on the fifth row of Figure 4.15 (top fin of top fish). The artifacts are more noticeable in the approximation that uses five images. As in the dice light field, the region where the artifacts are most visible, seem to be the regions where intensity changes from pure white to pure black (resulting in a large magnitude derivative value). Ghosting and blocking artifacts appear along the main edges of the images in the **bunnies** light field. The performance on the light field **TRex** is


Exact
One image
Five images
1
Image (3,5)
Image (3,5)
Image (3,5)
2
Image (3,5)
Image (3,5)
Image (3,5)
3
Image (1,1)
Image (1,1)
Image (1,1)
4
Image (1,1)
Image (1,1)
Image (1,1)

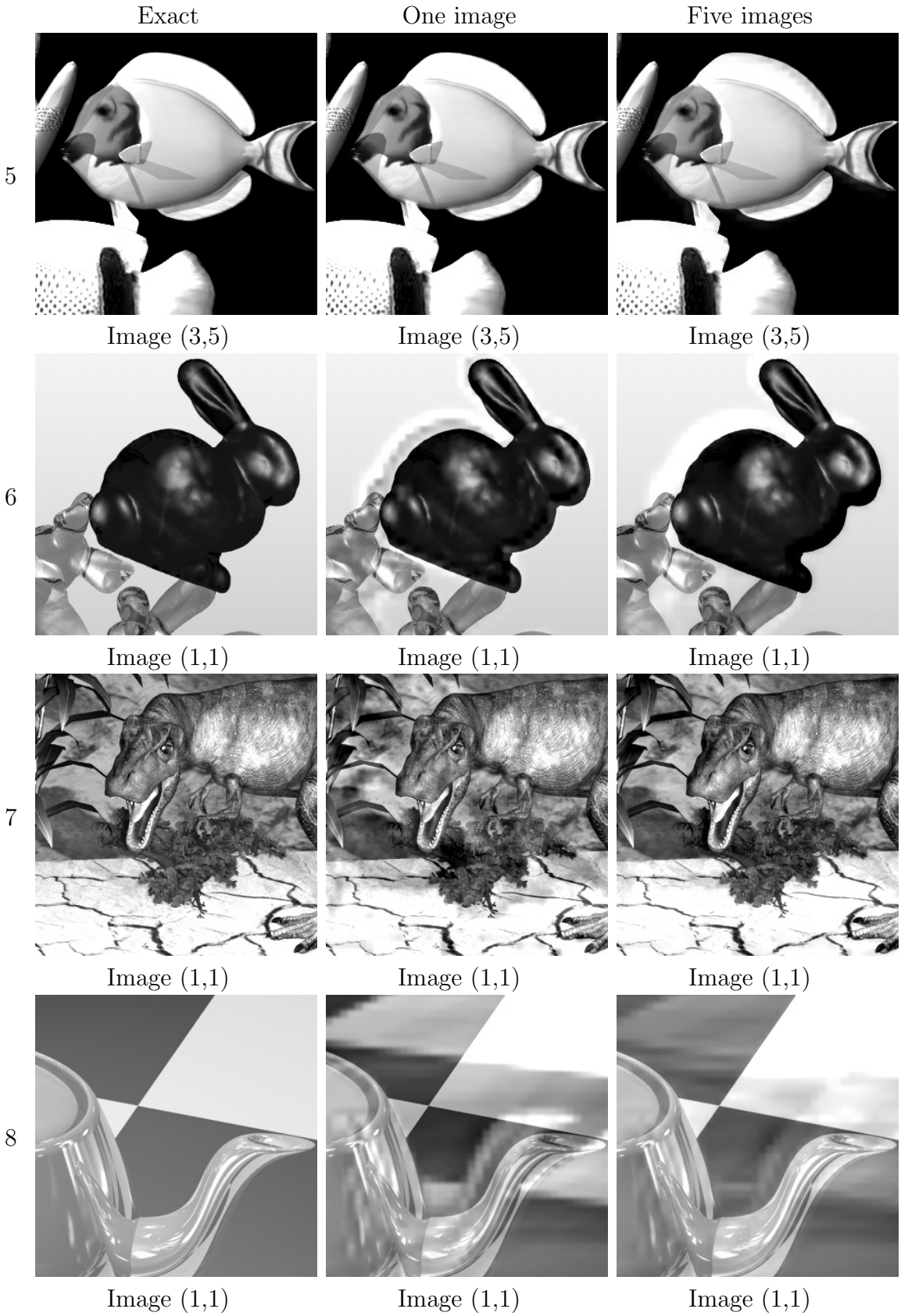


Figure 4.15: Cropped regions from 5×5 arrays of images. Regions and images selected to highlight the artifacts introduced by approximating $\hat{\Phi}^0_{LLLL}$.

Table 4.2: Light fields with a 7×7 array structure from [1]

Light field ID	Light field name	Individual image size
1	Butterfly	384×512
2	Red dragon	
3	Green dragon	
4	Happy Buddha	
5	Lucy	
6	Mini Cooper	

particularly good and if one would not be shown the original data set, most likely they would not perceive any artifacts in the reconstructed versions.

The consequences of the approximation are particularly visible on the light field **teapot**. Using one or five images to generate the approximation of the missing entry $\hat{\Phi}^0_{LLLL}$ generates ghosting as well as blocking around the main edges. This poor performance can be explained by the fact that this light field has a very strong motion parallax, resulting in a relatively low similarity between the images in the light field. This is illustrated in Figure 4.13, where SSIM maps are shown for the **teapot** and **Messerschmitt** light fields, on which the effects of the approximation were less noticeable. Lower values of SSIM indicate less similarity across the images in the light field.

4.6.2.2 Results on 7×7 arrays of images

This dataset is also computer generated and was downloaded from the same source [1]. The field of view is wide, as they too were developed with applications such as glasses free displays in mind.

On this dataset the approximation using five images consistently scores better than the approximation using one image, in terms of both average PSNR and SSIM, as can be seen from Figure 4.17.

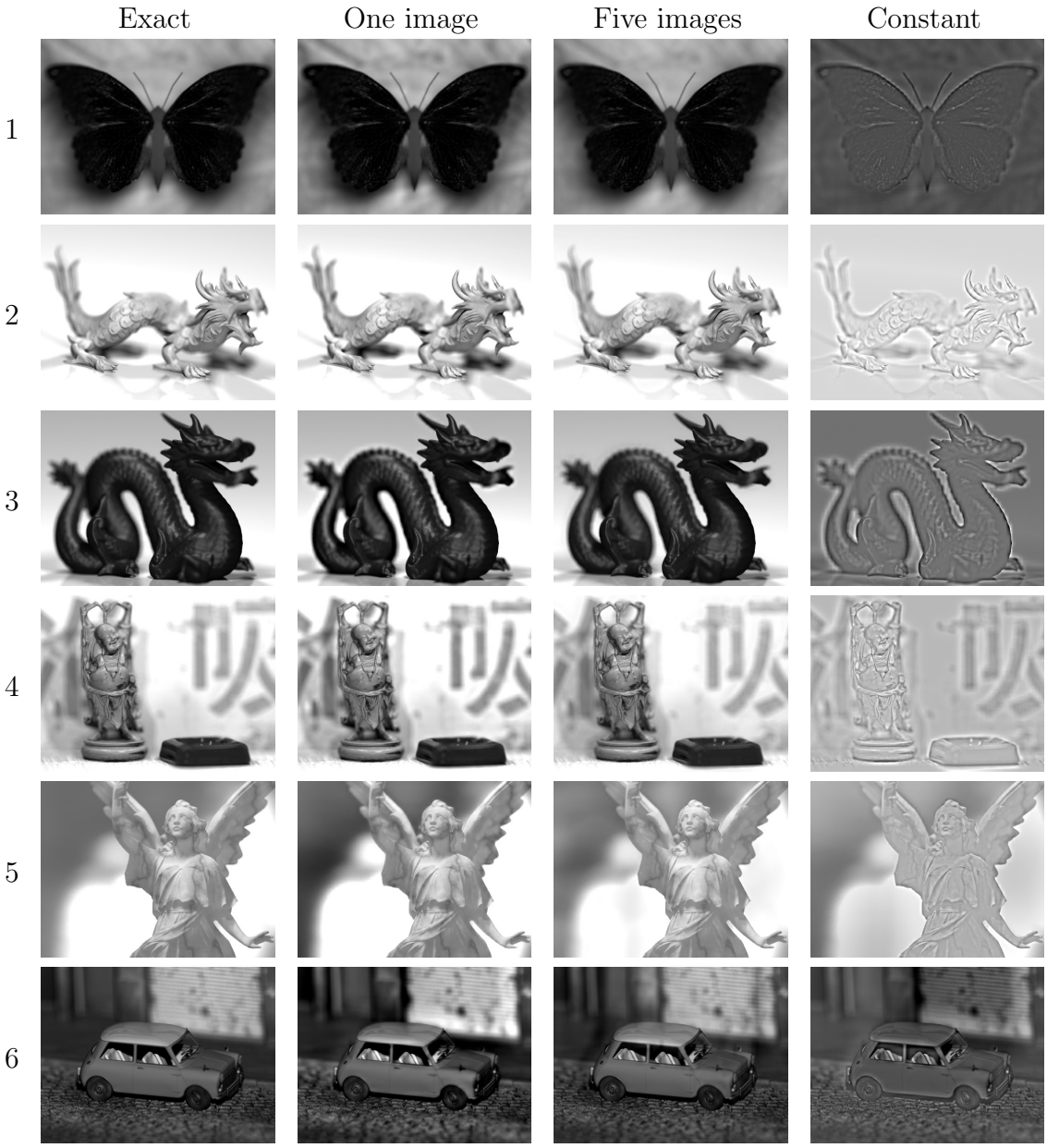
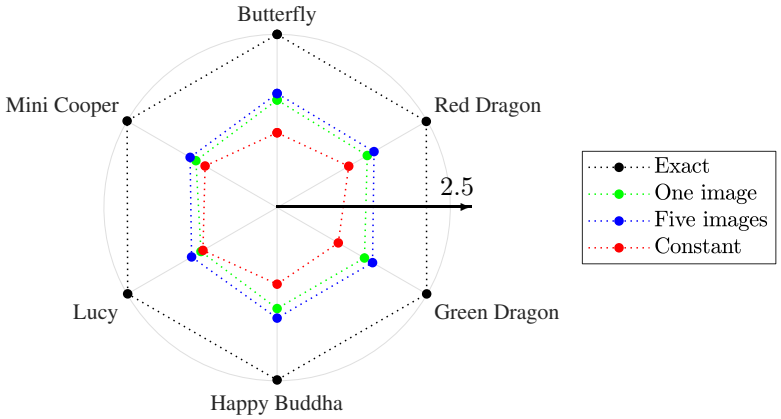
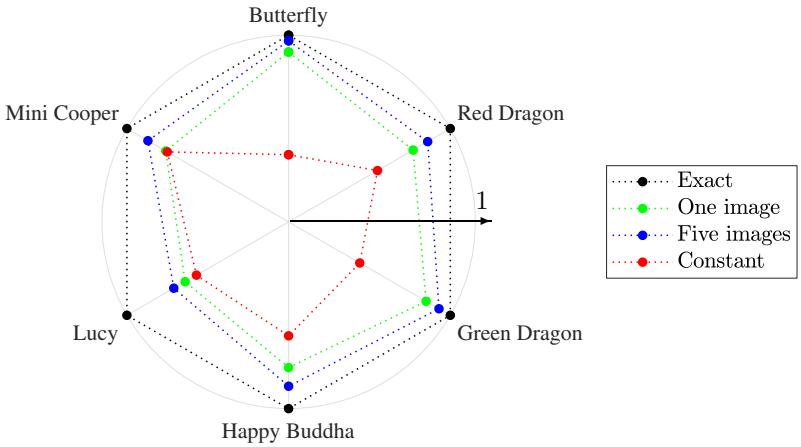


Figure 4.16: Central views of light fields listed in Table 4.2. Results from left to right on each row show the central views of the following light fields: reconstructed from the derivatives, and taking the $\hat{\Phi}^0_{LLLL}$ from the wavelet decomposition of the light field; reconstructed from the derivatives, using the gradient of only one image in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, using the gradient of five images in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, approximating all entries of $\hat{\Phi}^0_{LLLL}$ with a constant.



(a) Logarithmic representation of average PSNR values of reconstructed versus original light fields. The values farthest from the center denote a better quality of the reconstructed images.



(b) Linear representation of average SSIM values between reconstructed and original light fields. Values closest to 1 indicate better quality of the reconstructed images.

Figure 4.17: PSNR and SSIM values of all images in the light fields listed in Table 4.2, reconstructed from their gradient, compared against their original counterparts from database [1].

4.6.2.3 Results on 9×9 arrays of images

4.6.2.3.1 Lytro dataset

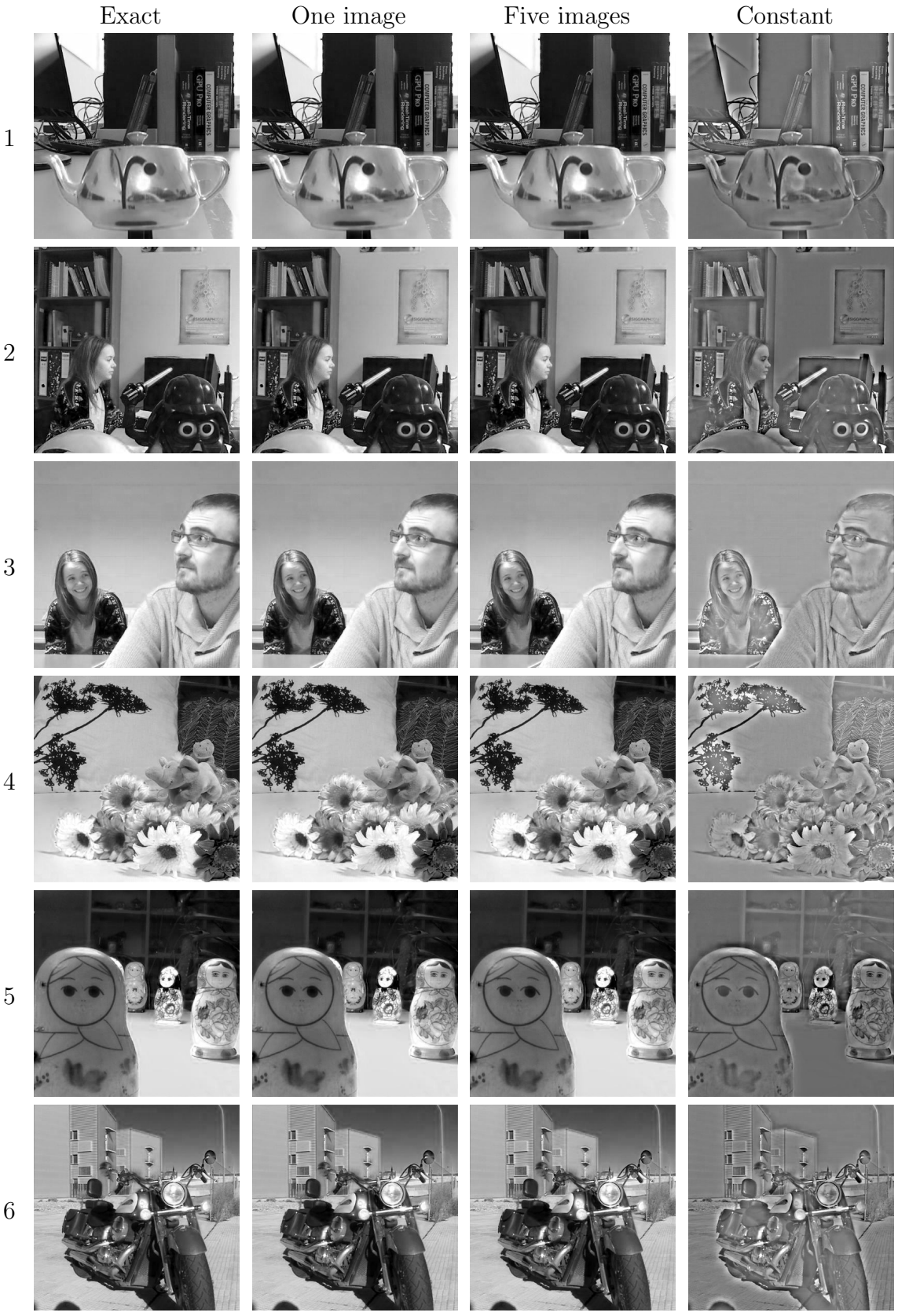
The Lytro dataset [2] consists of images of real scenes, taken with a Lytro camera (first version). The spatial resolution of all images in the array is 400×400.

On the Lytro dataset [2], the approximation using one image yields reconstructions very close in quality to the approximation produced by using five images, in terms of both average PSNR and SSIM (see Fig. 4.20). Both approximations yield very good results compared to the exact reconstruction, in terms of average SSIM, as seen in Fig. 4.20b.

The overall performance of the algorithm is good on this dataset, as is illustrated in the detailed crops shown in Figure 4.19. Here, zoomed in regions with size 150×150 are shown, to help visualize the effects of different types of approximations on the reconstructed light fields. The light fields with the best visual quality are **computer**, **couple** and **frog**. Some ghosting effects are noticed in the light field **matrioska** when the missing subband entry is approximated from five images in the light field. Minimal artifacts are perceived mainly as slight shading variations in the light fields **cg**, **motorbike** and **toys**, particularly when the missing subband entry is approximated from five images in the light field.

Table 4.3: Light fields with a 9×9 array structure from [2]

Light field ID	Light field name	Individual image size
1	CG	400×400
2	Computer	
3	Couple	
4	Frog	
5	Matrioska	
6	Motorbike	
7	Toys	


Exact
One image
Five images
Constant
1
2
3
4
5
6

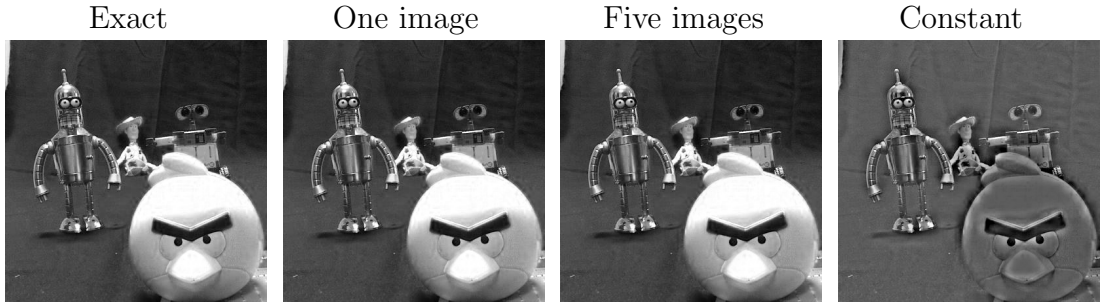


Figure 4.18: Central views of light fields listed in Table 4.3. Results from left to right on each row show the central views of the following light fields: reconstructed from the derivatives, and taking the $\hat{\Phi}^0_{LLLL}$ from the wavelet decomposition of the light field; reconstructed from the derivatives, using the gradient of only one image in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, using the gradient of five images in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, approximating all entries of $\hat{\Phi}^0_{LLLL}$ with a constant.

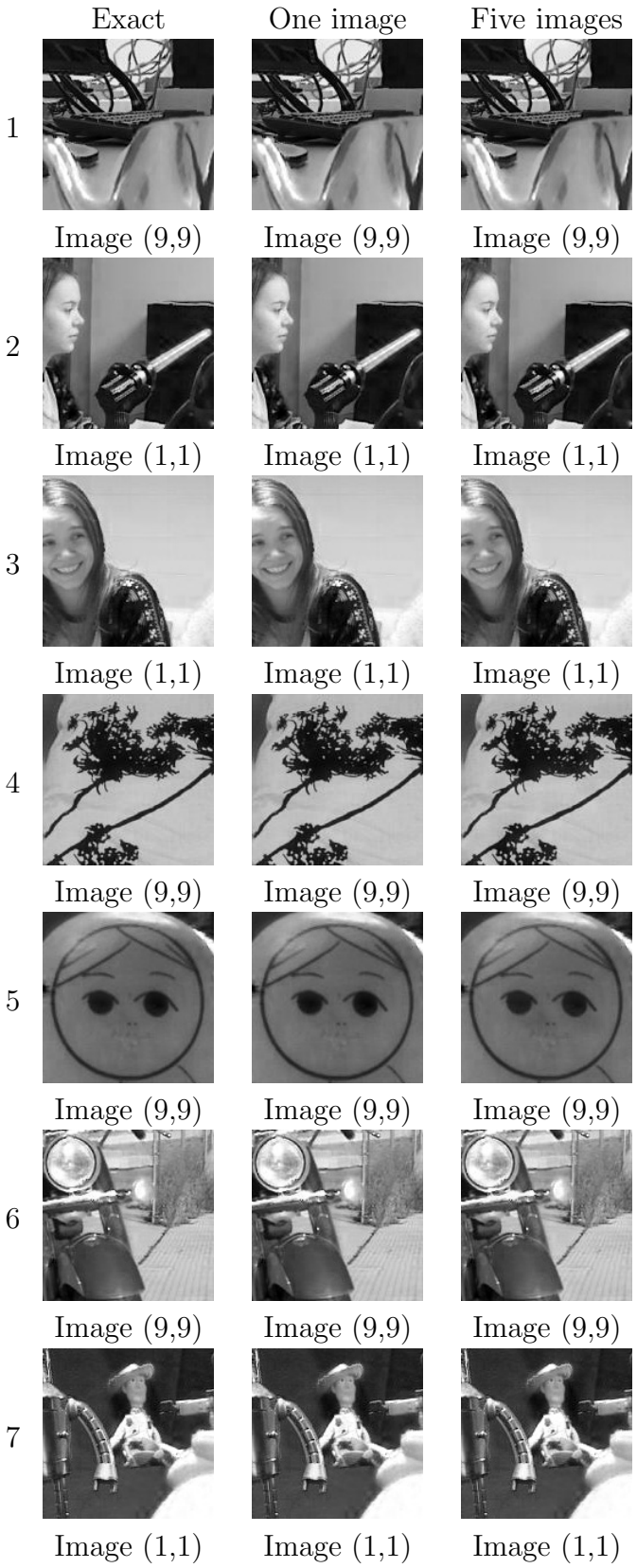
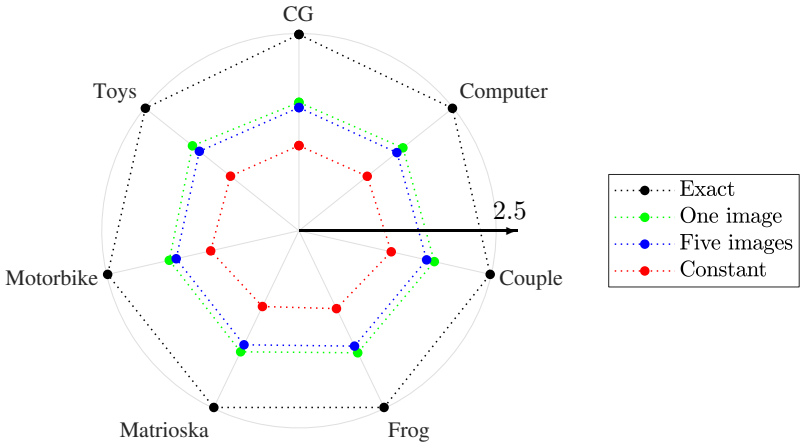
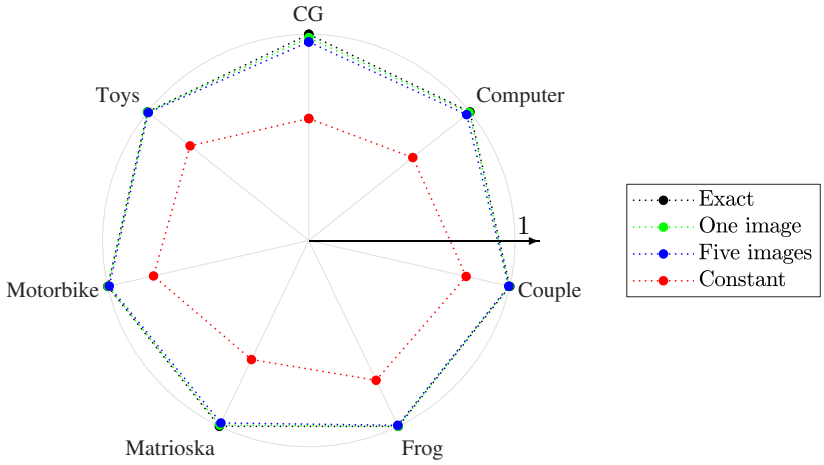


Figure 4.19: Cropped regions from 9×9 arrays of images. Regions and images selected to highlight the artifacts introduced by approximating $\hat{\Phi}^0_{LLLL}$.



(a) Logarithmic representation of average PSNR values of reconstructed versus original light fields. The values farthest from the center denote a better quality of the reconstructed images.



(b) Linear representation of average SSIM values between reconstructed and original light fields. Values closest to 1 indicate better quality of the reconstructed images.

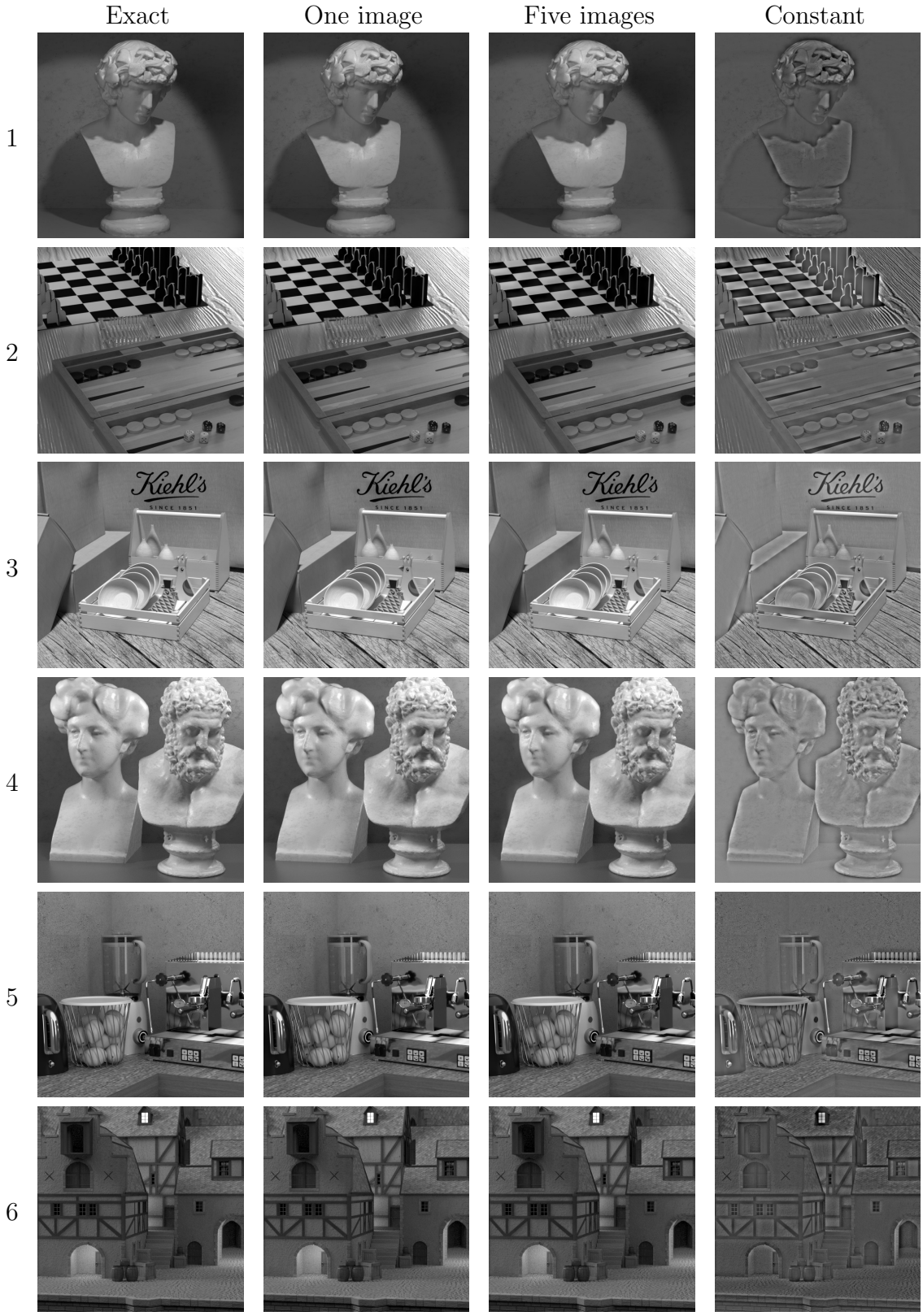
Figure 4.20: Algorithm performance on 9×9 arrays from the Lytro database [2]. PSNR and SSIM values of all images in the light fields listed in Table 4.3, reconstructed from their gradient, compared against their original counterparts.

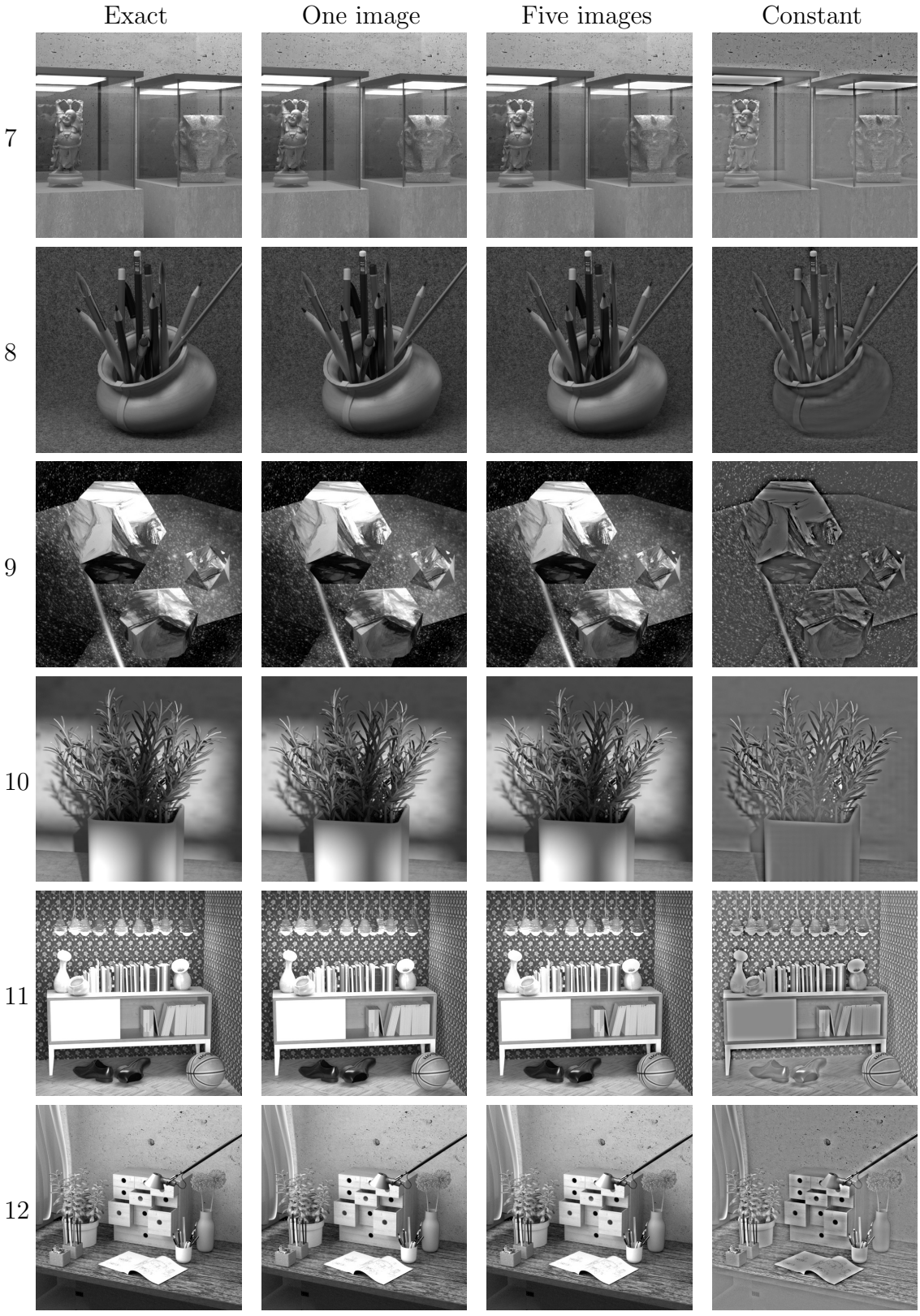
4.6.2.3.2 4-D Light field benchmark dataset

On the 4-D Light field benchmark dataset [3], the approximation using one image yields reconstructions very close in quality to the approximation produced by using five images, in terms of both average PSNR and SSIM (see Fig. 4.22). Both approximations yield very good results compared to the exact reconstruction, in terms of average SSIM (see Fig. 4.22b).

Table 4.4: 4-D light field benchmark database [3]

Light field ID	Light field name	Individual image size
1	Antinous	512×512
2	Boardgames	
3	Dishes	
4	Greek	
5	Kitchen	
6	Medieval	
7	Museum	
8	Pens	
9	Platonic	
10	Rosemary	
11	Sideboard	
12	Table	
13	Tomb	
14	Tower	
15	Tomb	
16	Vinyl	


Exact
One image
Five images
Constant
1
2
3
4
5
6
Kiehl's
SINCE 1851


Exact
One image
Five images
Constant
7
8
9
10
11
12

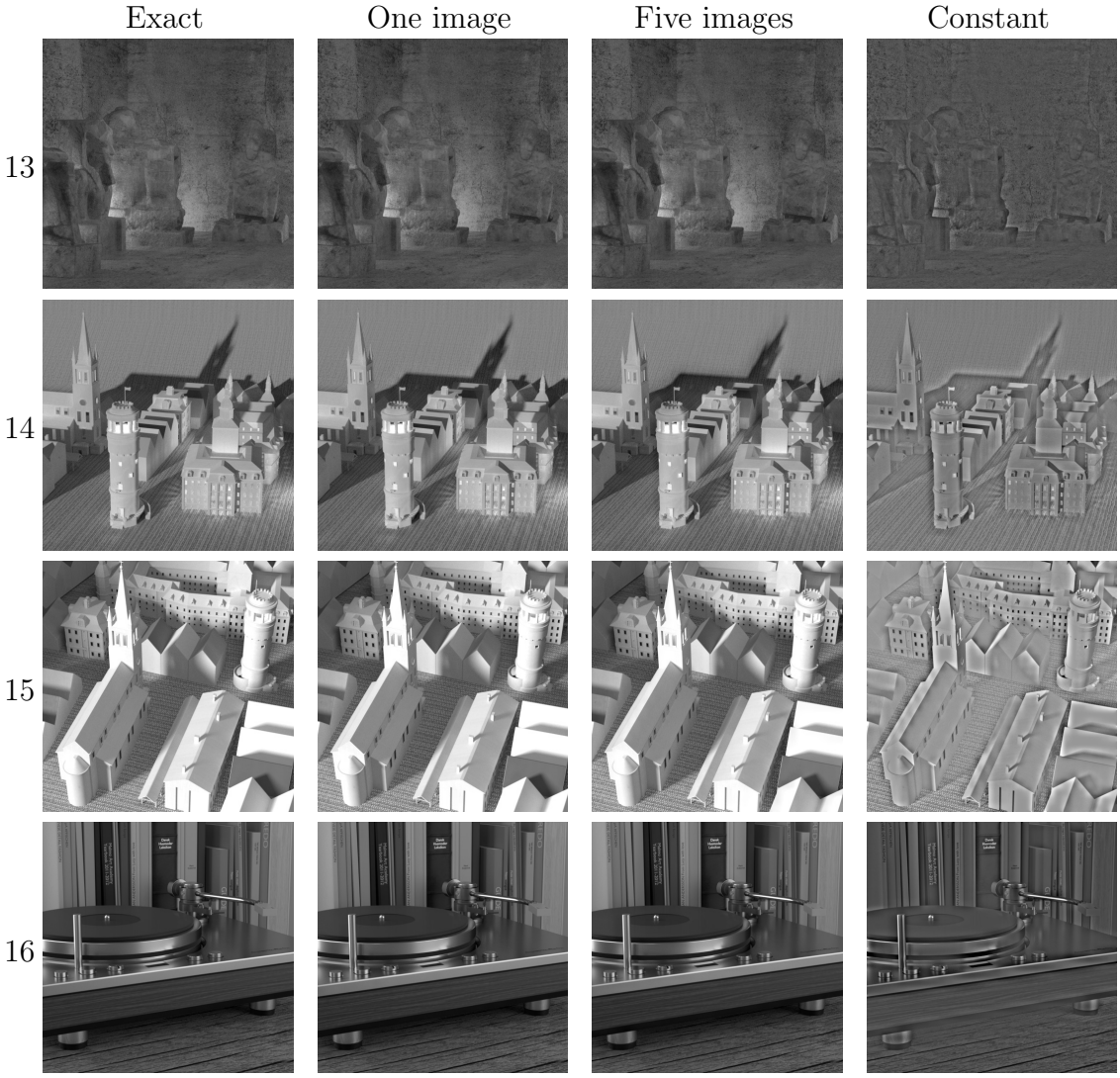
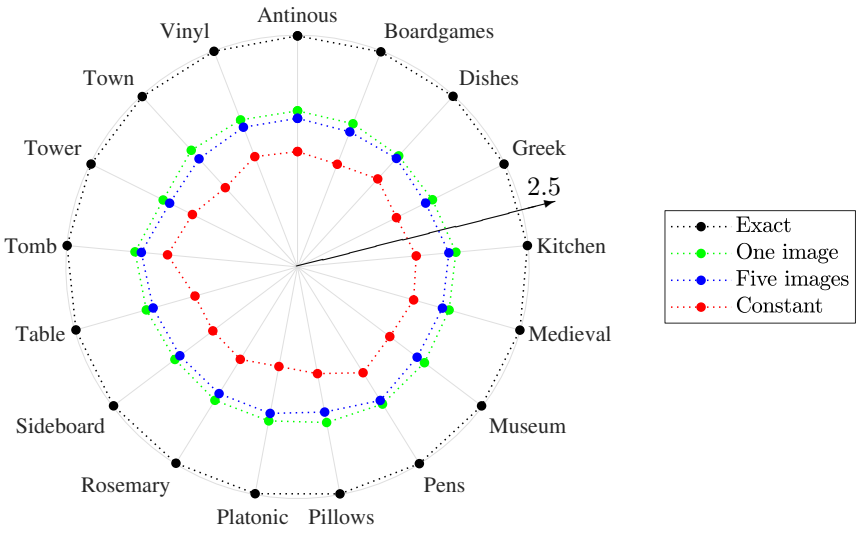
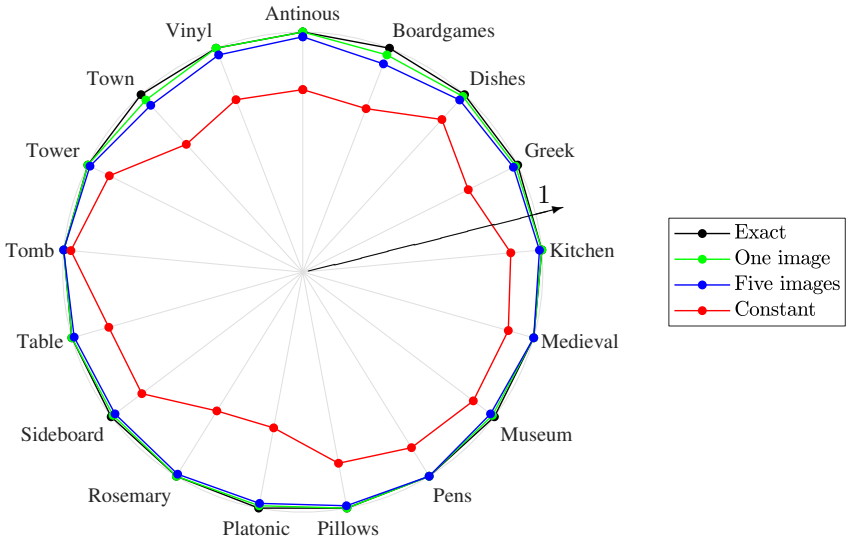


Figure 4.21: Central views of light fields listed in Table 4.4. Results from left to right on each row show the central views of the following light fields: reconstructed from the derivatives, and taking the $\hat{\Phi}^0_{LLLL}$ from the wavelet decomposition of the light field; reconstructed from the derivatives, using the gradient of only one image in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, using the gradient of five images in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, approximating all entries of $\hat{\Phi}^0_{LLLL}$ with a constant.



(a) Logarithmic representation of average PSNR values of reconstructed versus original light fields. The values farthest from the center denote a better quality of the reconstructed images.



(b) Linear representation of average SSIM values between reconstructed and original light fields. Values closest to 1 indicate better quality of the reconstructed images.

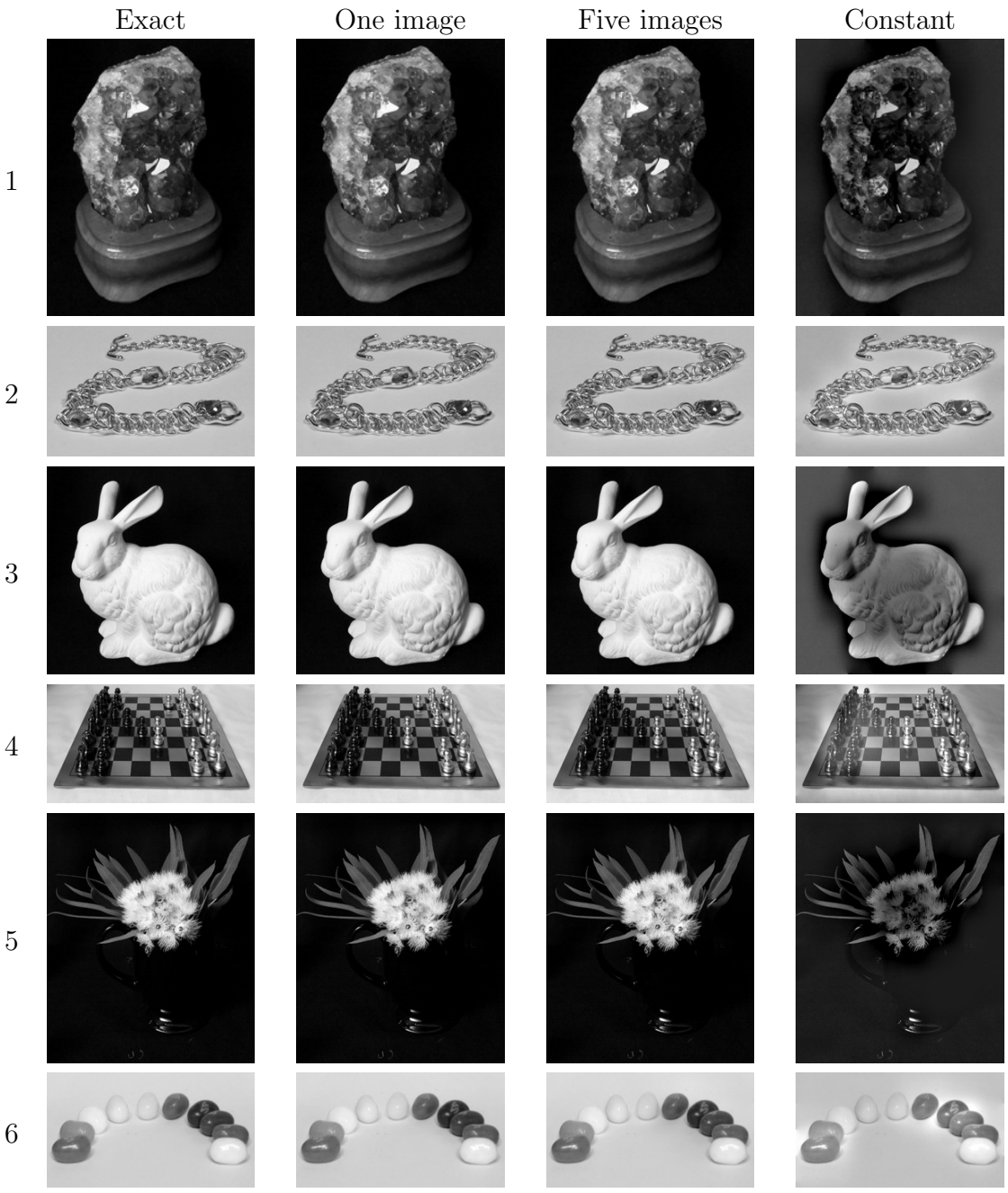
Figure 4.22: Algorithm performance on 9×9 arrays [3]. PSNR and SSIM values of all images in the light fields listed in Table 4.4, reconstructed from their gradient, compared against their original counterparts from database [3].

4.6.2.4 Results on 17×17 arrays of images

On this dataset, the approximation using one image consistently outperforms the approximation using five images, in terms of both analyzed indices. Note also the reasonable quality (SSIM) of the results produced by the simplest reconstruction type (approximating Φ^0_{LLLL} with a constant) on light fields **bracelet**, **chess**, **jellybeans** and **tarot**. This is explained by the fact that these light field contain quite a bit of details, and that information is not encoded in Φ^0_{LLLL}, but rather in the rest of the coefficients of the wavelet decomposition, successfully recovered by the algorithm from the light field derivatives.

Table 4.5: Light fields in Stanford database [4]

Light field ID	Light field name	Individual image size
1	Amethyst	256×192
2	Bracelet	160×256
3	Bunny	256×256
4	Chess	146×256
5	Eucalyptus	307×256
6	Jellybeans	141×256
7	Tarot	256×256


Exact
One image
Five images
Constant
1
2
3
4
5
6

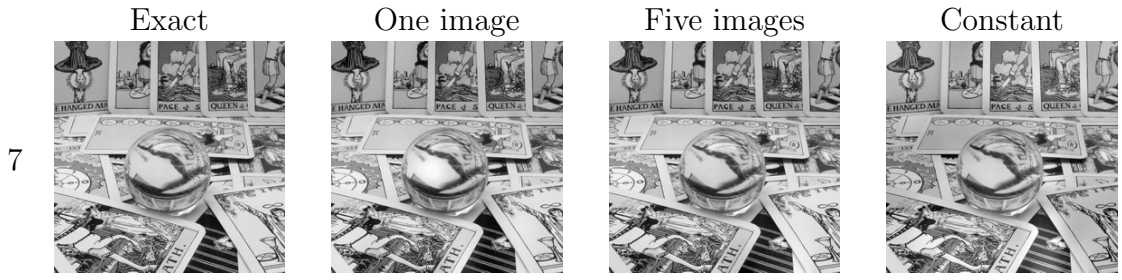
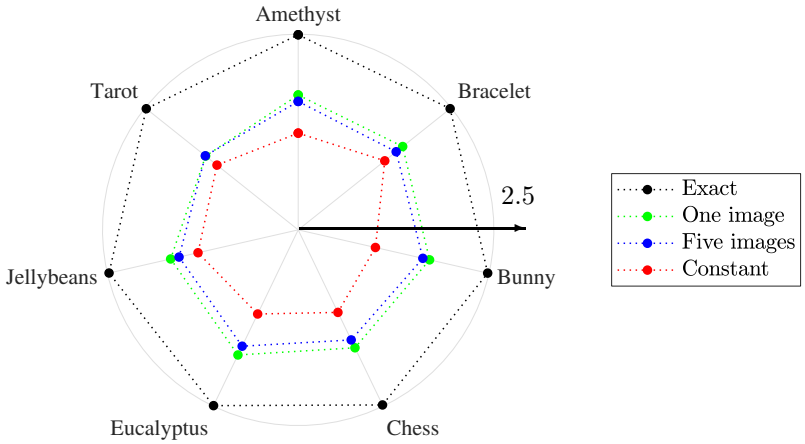
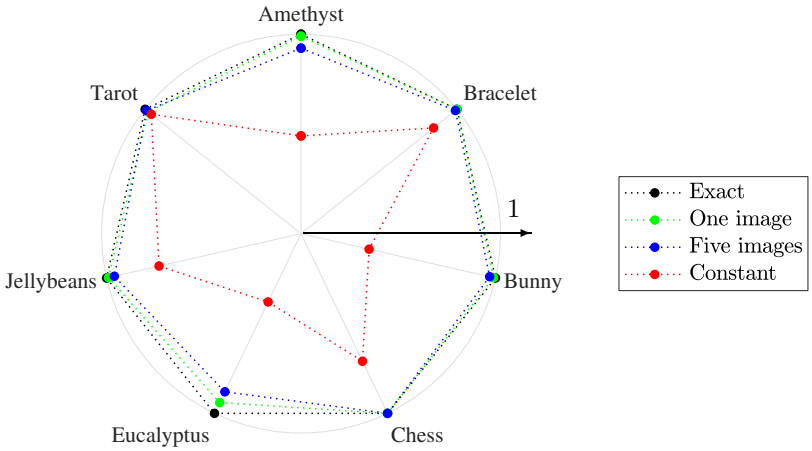


Figure 4.23: Central views of light fields listed in Table 4.5. Results from left to right on each row show the central views of the following light fields: reconstructed from the derivatives, and taking the $\hat{\Phi}^0_{LLLL}$ from the wavelet decomposition of the light field; reconstructed from the derivatives, using the gradient of only one image in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, using the gradient of five images in the light field to approximate $\hat{\Phi}^0_{LLLL}$; reconstructed from the derivatives, approximating all entries of $\hat{\Phi}^0_{LLLL}$ with a constant.



(a) Logarithmic representation of average PSNR values of reconstructed versus original light fields. The values farthest from the center denote a better quality of the reconstructed images.



(b) Linear representation of average SSIM values between reconstructed and original light fields. Values closest to 1 indicate better quality of the reconstructed images.

Figure 4.24: Algorithm performance on 17×17 arrays. PSNR and SSIM values of all images in the light fields listed in Table 4.5, reconstructed from their gradient, compared against their original counterparts from database [4]

.

4.7 Conclusions

In this chapter, a technique to reconstruct a 4-D signal from a given gradient data set was proposed. The technique was developed specifically for applications where the signal of interest has a light field structure. The technique makes use of the separability property of the 4-D wavelet transform and of the similarity between the multiple views of the imaged scene. As an intermediate step, the Haar wavelet decomposition of the signal is obtained from the signal derivatives. All detail subbands of the complete Haar wavelet decomposition can be found from the signal derivatives, and a technique is also described for obtaining the lowest resolution approximation subband coefficients. Once the wavelet decomposition of the signal is obtained, the signal can be reconstructed from the wavelet decomposition via wavelet synthesis, with the option of including an iterative Poisson solver, a process referred here as modified wavelet synthesis.

The technique was then tested on five well-known light field datasets, with the intention of identifying its strengths and weaknesses. An important finding is that the best results were obtained when the disparity between the different views in the light fields was not too high, a quality intuitively depicted by SSIM maps of all images in the array, versus the central image in the array. Overall, the results show that the proposed algorithm works well to reconstruct both artificially generated and real light field datasets from a given gradient dataset. The results reported indicate the promising potential of the reconstruction tool developed in this chapter in light field applications, and this will be the object of the following chapter of this dissertation.

Chapter 5

Light field applications

5.1 Chapter outline

This chapter focuses on devising new algorithms using the technique developed in Chapter 4 for two well known image processing problems: image editing and image fusion. Unlike conventional approaches, where images are depicted as 2-D digital signals, this chapter approaches these problems in the 4-D light field domain. The reconstruction tool developed in Chapter 4 is integrated in two gradient based light field editing applications, illustrated with examples.

The chapter begins with the motivation for studying light fields applications. An overview of current state-of-the-art light field applications is provided in Section 5.3. The chapter continues with two gradient based light field applications, illustrated with examples in Sections 5.4 and 5.5. An discussion about light field quality evaluation is presented in Section 5.6. Finally, Section 5.7 summarizes the developments of this chapter and provides recommendations for future work.

5.2 Motivation

Light field imaging offers more comprehensive scene representations than those produced by traditional imaging devices and techniques [68]. The richness of information that light fields contain comes unsurprisingly with its share of tradeoffs, that begins from devising practical means to acquire and store light fields, and continues to designing displays capable to render them in a meaningful and intuitive manner. Last but not least, it is exciting to revisit well-known image processing problems and use

the richness of information embedded in light fields scene representations to explore the possibility of developing better quality solutions. The most recent applications that have light field as signal of interest are reviewed in what follows.

5.3 Current light field applications

Being able to choose which areas are in focus and which are not *after* an image has been recorded is a desired feature in many applications. If the image is recorded with a conventional camera (such as a compact or digital single reflex camera), this can be attempted either by deblurring algorithms, to bring in focus certain regions in the image – a notoriously challenging task – or by filtering, to simulate out of focus areas. Refocusing becomes a simpler task if the image is recorded as a 4-D light field [69]. This in turn has interesting consequences in robotics [62], [70] or security [56], [60], where it enables cameras to detect objects of interest situated behind occlusions.

Another interesting light field application is novel view synthesis, i.e. the capability of generating a new two dimensional view of an imaged scene from a given light field. This was studied in [71], who introduce LightShop, one of the first systems that allows users to "interactively manipulate, composite and render multiple light fields" [71]. Kubota et al. [72] propose synthesizing a novel view of a scene by generating a light field from two sets of differently focused images taken by an aperture camera array. The method is designed for scenes with clearly separated layers.

Recognition algorithms also benefited from the richness of information contained in light fields scene representations. Gross et al. [73], [74] use light-fields to develop an appearance-based face recognition algorithm (preprocessing data using PCA and classification using nearest neighbour), robust to pose variation. Raghavendra et al. [75] study the face recognition problem on a dataset of face images that they acquire using both a conventional and a light-field camera. They use the energy of the Haar wavelet transform to select the best focused face image from the light field and use this as input for the recognition algorithm. They report higher recognition rates for the images acquired with the light field camera than for those acquired with the conventional camera, despite the low spatial resolution of the light field images.

Wang et al. [76] study the notoriously challenging problem of image separation, with the purpose of eliminating glass reflections. They develop a technique that uses light field information to approximate the depth of the scene of the desired layer. Using the light field camera provides their algorithm with a scene representation

consisting of a set of images from different viewpoints, often required in conventional image separation algorithms. The benefit is that this scene representation is obtained at the same time, using the same imaging device, and this eliminates the need for user interaction. Johansen et al. [77] develop a light field based image separation technique based on sparse coding, while Li and Lun [78] look at the same problem, and develop a technique that uses information contained in the gradients of the light field epipolar plane image to perform the separation task.

Mukati et al. [79] use a first-generation Lytro [59] to study the problem of multiple light field stitching. Their technique is based on identifying a transformation that is used to register corresponding sub-aperture images in the two light fields. This transform consists of a rotation, scale and translation; the scale and translation between light fields are determined using the central images in the light fields. Birklbauer and Bimber [80], [81] put forth a novel approach to the light field stitching problem. They use a light field camera to record overlapping "sub-light-fields" scene representations, with an a priori known camera motion model, and generate a panoramic light field, without the need for depth reconstruction or image registration. Guo et al. [82] also propose a light field stitching technique which uses pairwise ray-ray correspondences for the registration part and multi-resolution, high-dimensional graph-cuts for the stitching, being therefore also suitable to stitch non-static light fields. Their technique does not require explicit knowledge of camera motion.

Srinivasan et al. [83] study the effects of light field camera motion on the properties of the captured light field and propose a blind deblurring method based on "modulating a slice of the Fourier spectrum of the motion-blurred light field" [83]. A comparison between the technique they develop and state-of-the-art techniques that deblur individual subaperture images of the light field shows that applying 2D deblurring algorithms on subaperture images causes inconsistency between neighbouring views in the light field.

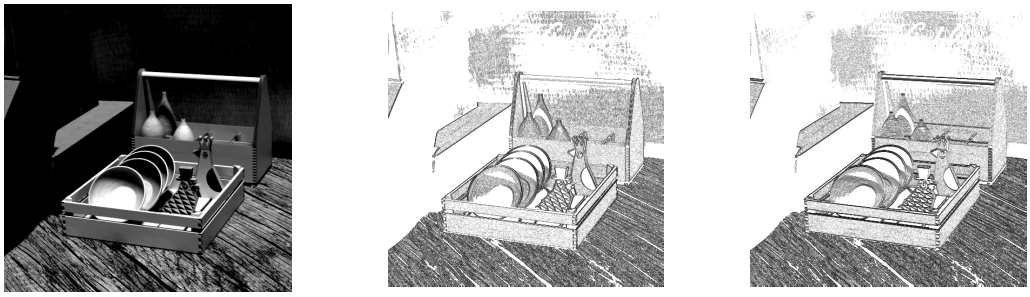
5.4 Multi exposure light field fusion

The only work that deals with light field fusion is developed in the field of microscopy [84], where the authors develop a lensfree microscopy system with the goal of producing high contrast, high resolution images of biological specimens. The light field notion used here is different than the one used in the computer vision and image processing communities, as it makes use of the wavelegth dimension of light, which

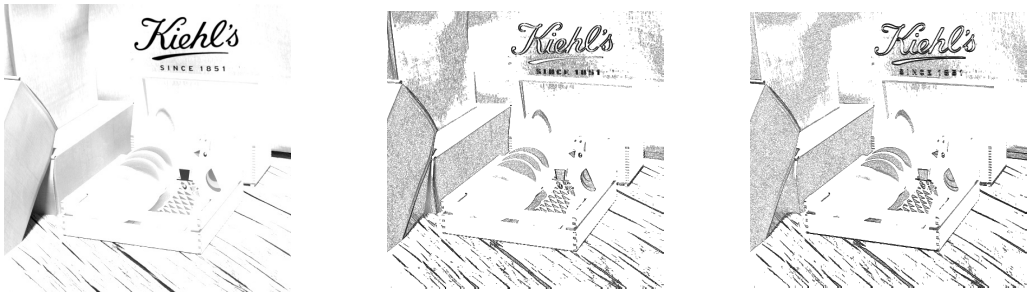
is typically disregarded in the studies originated in these communities.

In this section, the problem of light field fusion is investigated, in the context of image processing. In particular, the focus is on developing a gradient based technique of fusing together over and under exposed light fields, with the algorithm developed in Section 2.7.2 as a starting point.

A light field called *dishes* from database [3] is used in this example. The size of the original light field is $9 \times 9 \times 512 \times 512$. All subaperture images were consistently modified using an external image processing software, to simulate highly underexposed and overexposed light field representations of the same scene. In Figure 5.1 the central views and the spatial gradient components of the luminance channel of these two light fields are shown. As can be observed from Figures 5.1b and 5.1d, certain details, such as the writing, are largely lost in the underexposed version of the light field, while others, such as the dishes contours are lost in the overexposed version.



(a) Underexposed view **(b)** Gradient components



(c) Overexposed view **(d)** Gradient components

Figure 5.1: Central views of artificially generated under-exposed and over exposed light fields. Original light field from [3].

First, the gradients of the two light fields are computed and used as input data for the fusion algorithm. Then, the magnitudes of the two gradients are computed and compared to generate a 4-D index map of the locations of the largest components. Points situated at the locations corresponding to the largest magnitude gradient are

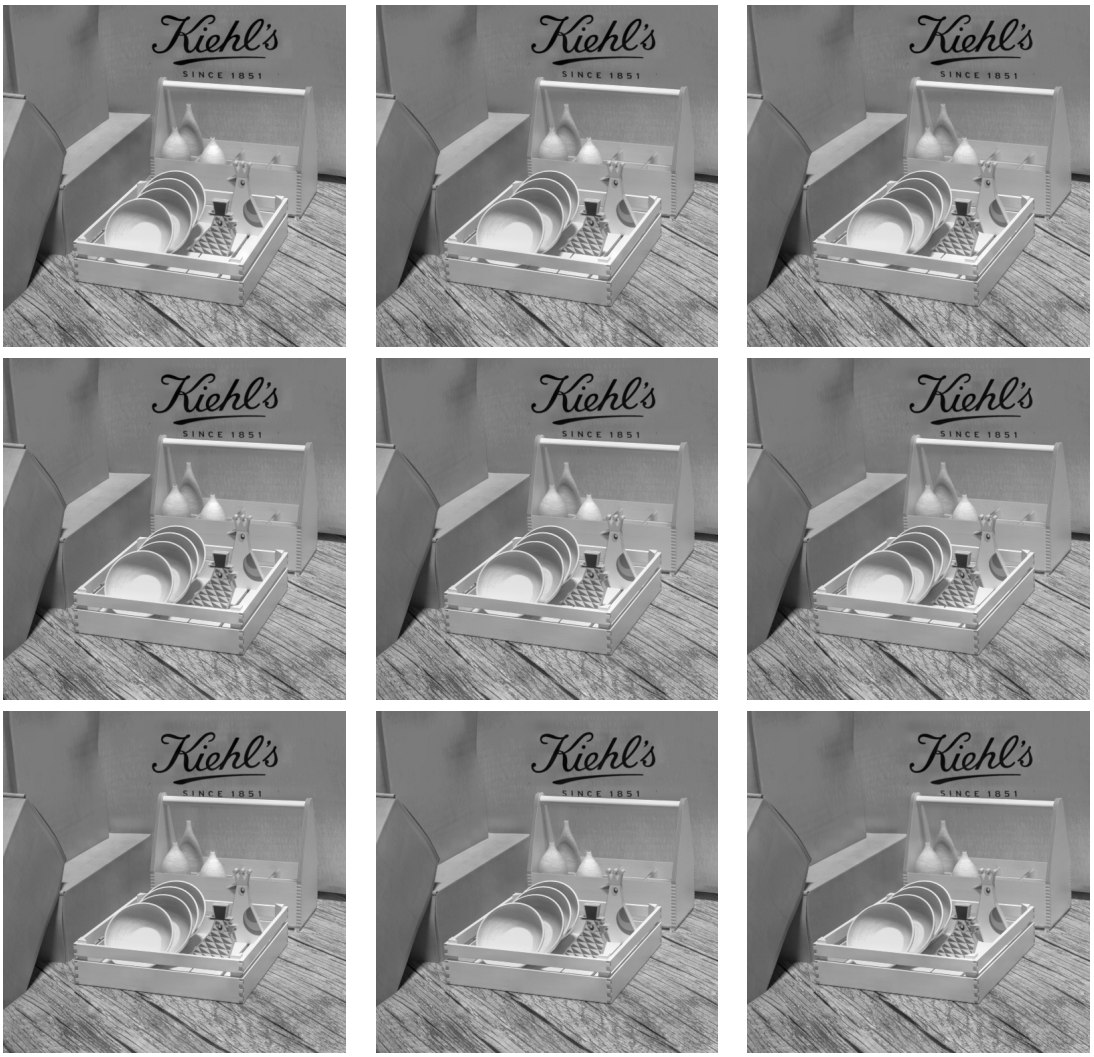


Figure 5.2: Select views from fused light field

selected from the gradient of each light field to generate the gradient of the fused light field. Finally, the fused light field is obtained from this gradient data set by using the reconstruction algorithm developed in Chapter 4. The fused light field is illustrated by nine representative views in Figure 5.2. As the figure shows, details that were lost in the underexposed or overexposed versions of the light field were successfully recovered. The algorithm also maintains inter-view consistency.

5.5 Light field editing

The task of light field editing is new, and is far from simple, given the multi-dimensional nature of the signals of interest. Recent efforts [2], [66] focus on understanding users' preferred modes of interacting with light field scene representations, as well as possible workflows. In [2], users have access to a multiview interface and a multifocus interface, with the purpose of performing simple editing tasks on a given light field (e.g. changing color or brightness of a certain surface or painting on an object). In the multiview interface, the user can select a specific subaperture image from the light field and is shown how the edit would propagate in the light field. In the multifocus interface, the user first selects a certain depth in the light field, and then performs the editing task. Later versions of these two interfaces [66] incorporate depth selectivity. The conclusion of this user studies is that the choice of interface is dictated by the task at hand.

An editing patch based framework called PlenoPatch is introduced by Zhang et al. [85]. The considered editing tasks are: removing an object from a given light field, modifying its depth, increasing the resolution of the light field and parallax magnification to increase the depth range of the scene. Their technique uses information from a 2-D image captured with a DSLR to increase the resolution of a light field captured with a Lytro [59] camera, provided that the two cameras are depicting the scene from the same location. The quality of their results depends on exact depth estimation.

In this work, the focus is on developing a solution to the problem of light field content replacement. The objective is to devise this solution without estimating the depth of objects in the scene, but rather by gradient domain manipulations.

5.5.1 Content replacement

A light field called *jellybeans* from database [4] is used in this example. A cropped version of the original light field with size $17 \times 17 \times 512 \times 928$ was obtained. All subaperture views of this light field were resized and a light field with size $17 \times 17 \times 141 \times 256$ was generated. The objective is to seamlessly insert the jellybeans from the smaller light field (referred to as *source light field*) in the larger version of the light field (*target light field*), at a given location, with minimal intra-view artifacts and no inter-view artifacts. The task is approached in the gradient domain. Each channel of

the color light field is processed individually and the process described next is applied to each of the three channels.

First, the four directional derivatives of the source light field are computed, and the gradient of the source light field is obtained.

$$\Phi_s = [\Phi_{s,u}, \Phi_{s,v}, \Phi_{s,x}, \Phi_{s,y}]^T \tag{5.1}$$

where directions u and v index the array of images in the multi view representation of the source light field and x and y index pixels in a view. Zero Neumann boundary conditions are assumed for all directional derivatives.

Next, the absolute values of the gradient components are added up and a magnitude map is obtained:

$$M = |\Phi_{s,u}| + |\Phi_{s,v}| + |\Phi_{s,x}| + |\Phi_{s,y}| \tag{5.2}$$

The size of this magnitude map is $17 \times 17 \times 141 \times 256$. Each of the 289 views of this magnitude map is then used to generate 289 binary masks, obtained by setting all values above a view specific threshold to 1 and setting all other values to 0. Let $M_{2D}(x, y)$ denote one of the 289 views of M. Each view M_{2D} is transformed to a binary mask b, using:

$$b(x, y) = \mathcal{T}(M(x, y)) \tag{5.3}$$

where:

$$\mathcal{T}(M_{2D}(i, j)) = \begin{cases} 1 & \text{for } M(i, j) \geq T \\ 0 & \text{for } M(i, j) < T \end{cases} \tag{5.4}$$

Each of the 289 matrices b are then placed in a 4-D array, which is used to weigh the gradients of the source and target light fields, and generate a gradient dataset from which the desired light field is to be obtained. The reconstruction algorithm developed in Chapter 4 is used to obtain the light field from the edited gradient data set. Three iterations of the Poisson solver are used at each resolution. Representative views of the edited light field are shown in Figure 5.3.

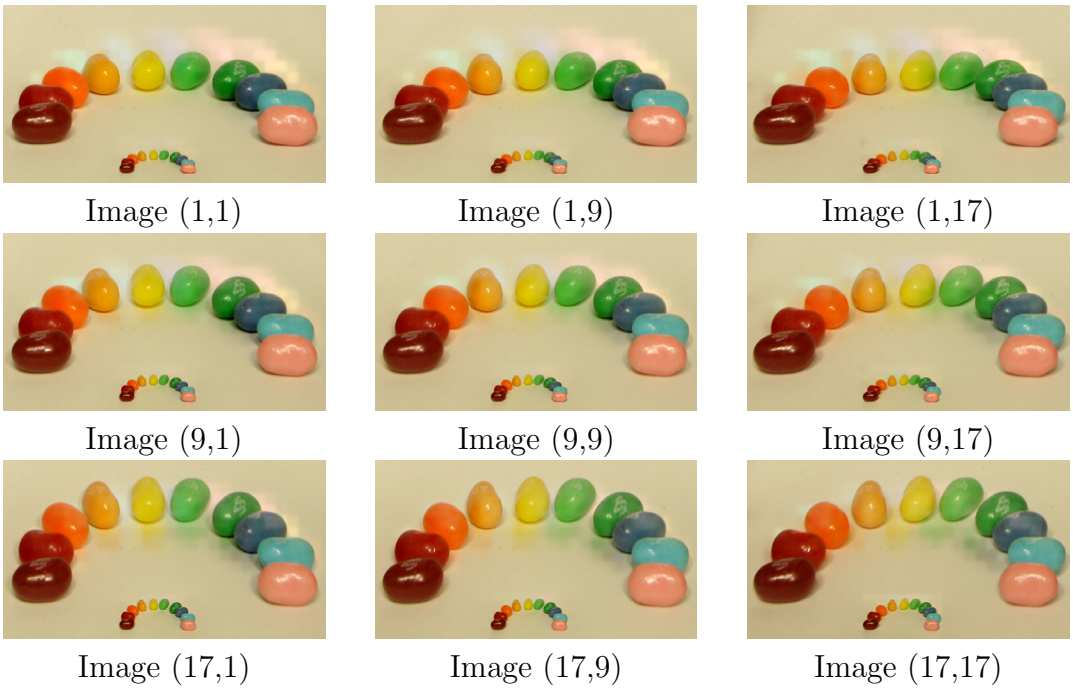


Figure 5.3: Light field editing results - example 1.

5.5.2 Transparency

The example presented in Figure 5.4 uses the same light field, but places the source light field in a different part of the target light field. The purpose of this example is to illustrate that effects such as eliminating occlusions and adding transparency can be obtained by using the proposed algorithm.

5.6 Quality evaluation analysis

Assessing the quality of a light field is an open problem. Fu et al. [68] present a study that compares conventional images and light field images, using an ISO testing standard. The ISO testing standard that their study relies on has since been revised. Vieira et al. [86], and Viola et al. [87], [88] both use PSNR to determine the objective quality of compressed light field images, with Viola et al. also organizing user studies to subjectively rank the light fields. Both compare individual images (subaperture images) in a reference light field the with individual images in a compressed version of the light field.

Assuming a reference light field is available, assessing the quality of a modified

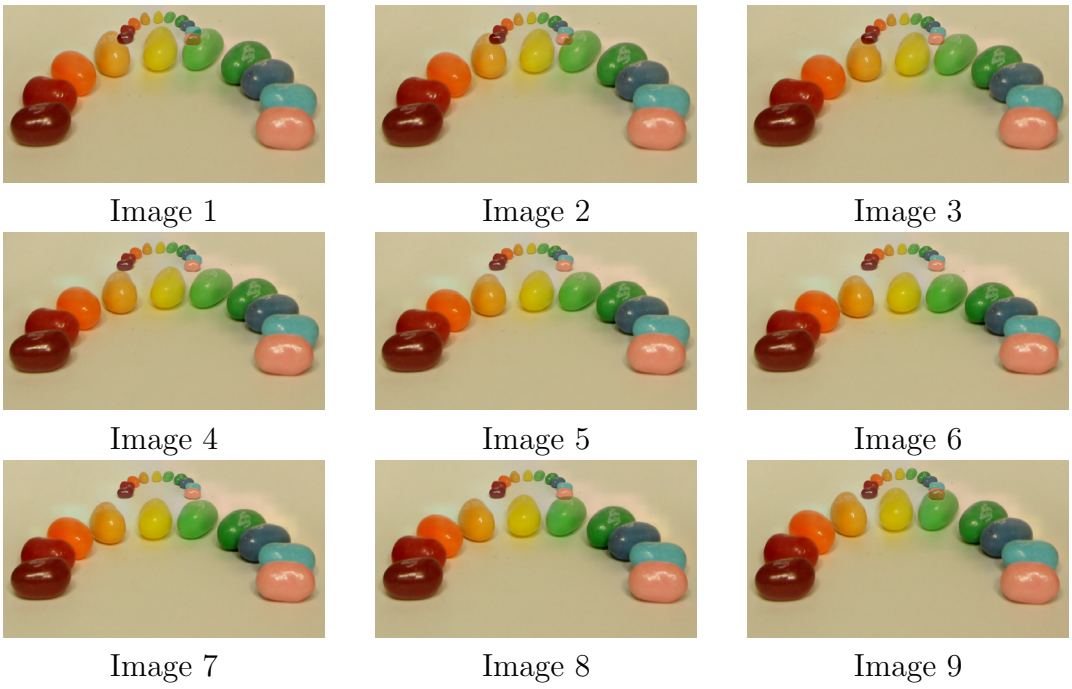
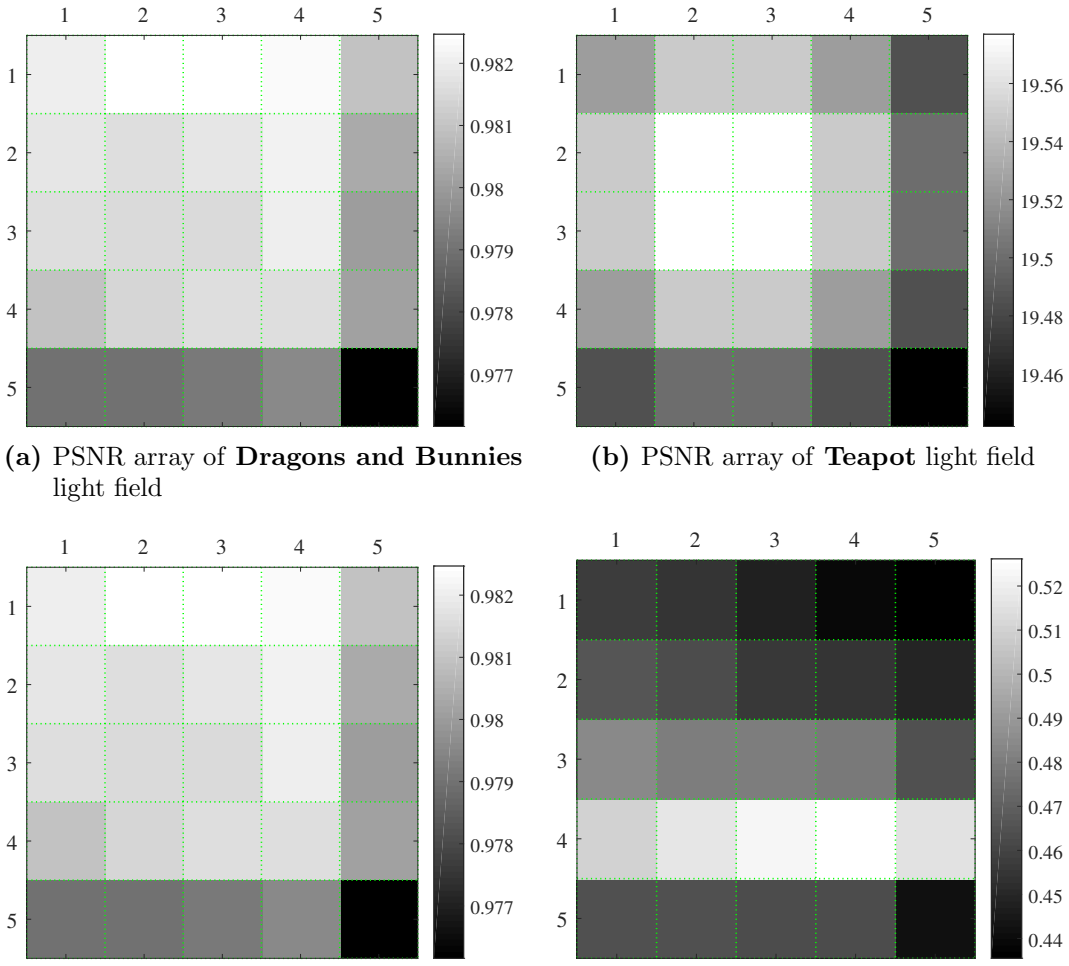


Figure 5.4: Light field editing results - example 2.

version of the given light field by applying an existing 2-D quality measure, such as PSNR, to determine the quality of each subaperture image of the modified light field is a reasonable starting point. However, aside from the well known limitations of the PSNR, such approaches do not take into account the 4-D nature of the light fields, failing to detect inconsistencies between neighbouring views. Two full reference metrics were used to assess the quality of the results in Chapter 4. The operating principle of the two metrics was the same, but the 2-D metrics they were based on differed. Given a modified version of a light field (in the case of Chapter 4, the modified light field was the light field reconstructed from its gradient), these metrics compare corresponding 2-D subaperture views and are recorded in a 2-D array. The 2-D array of metrics values has the same size as the array of subaperture views. An illustration of these metrics for two 5×5 arrays of images is shown in Figure 5.5.

A possible quality index, used to determine the quality of the reconstructed light fields with respect to the original light fields in Chapter 4, is given by the average value of all elements in these matrices. This is a good starting point for a quality metric, but it is not able to discern between a light field with overall poor quality of the subaperture views and a light field that has good quality subaperture views towards the center and some errors towards the edges. Also, this index may fail to



(a) PSNR array of **Dragons and Bunnies** light field

(b) PSNR array of **Teapot** light field

(c) SSIM array of **Dragons and Bunnies** light field

(d) SSIM array of **Teapot** light field

Figure 5.5: Metrics for intra-view quality assessment. Matrices depict pairwise comparisons between views of original and approximated versions of a given light field (approximation from five images considered in this example).

detect inconsistencies between neighbouring views. An index that assesses light field quality should thus also take into account the disparity between neighbouring views.

5.7 Conclusions

The two experiments presented in this chapter show that gradient domain light field processing is possible, provided that one has access to a means to recover the signal back from artificially generated gradient data sets. The visual quality of the results obtained indicate that the technique developed in Chapter 4 is a good candidate to perform the reconstruction task and demonstrates its value for other gradient based light field processing applications as well.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

In this dissertation, a tool is developed to efficiently reconstruct multi-dimensional digital signals with different dimensions from a given gradient data set. Developing this technique is largely motivated by applications that rely on operations performed on the signal gradient values, rather than on the signal values themselves. These applications require a way to efficiently recover a meaningful signal from an artificially generated gradient data set.

A secondary motivation, unexplored in this work, but worth mentioning nonetheless, is for applications such as adaptive optics, where the gradient of the signal of interest is measured by sensors, and again, a means to recover the signal itself is necessary for data analysis and interpretation. The gradient camera proposed in [89] is another application where gradient values are readily available. Studies concerning the robustness against noise of the proposed approach have been conducted in [54] and [16].

In Chapter 2, an algorithm to reconstruct 2-D signals from gradient data is proposed. The approach is based on an important observation put forth by Hampton et al. [15] for wavefront reconstruction in adaptive optics. Specifically, the algorithm is based on the relationship between the Haar wavelet analysis filters and the signal gradient, and has two main stages. First, an analysis stage, in which a decomposition similar to the Haar decomposition of the 2-D signal to be reconstructed is produced from the given gradient data. The analysis stage is followed by a synthesis stage, in which Haar wavelet synthesis is employed to produce an approximation of the 2-D

signal of interest.

What distinguishes the work in this dissertation from earlier developments is the gradient discretization model employed and, more importantly, the way in which signals with non-square support are handled. Specifically, instead of extending the support of the given gradient data to that of a square signal, from which the Haar wavelet decomposition is obtained, a technique was developed to generate a rectangular Haar like wavelet decomposition. The focus was on the coarsest resolution subband coefficients of the wavelet decomposition, and a means to find these coefficients from the gradient and the mean value of the signal was proposed. An intuition was also gained about the role that these coefficients have on the quality of the reconstructed signal, in the special case of digital images. A notable finding was that, in most cases when the signal of interest is a digital image, the effects of approximating these coefficients with a constant are visibly attenuated by using a very low number of iterations of a 2-D Poisson solver in the synthesis stage.

The importance of devising efficient ways to process signals with non-symmetric regions of support becomes apparent as we move forward to the realm of three dimensional signal processing. Expanding a signal with a non-cubic region of support to the nearest cube generates a large amount of data that is typically discarded after the signal is obtained. To avoid generating this additional data, an algorithm to reconstruct 3-D signals from a given gradient dataset was proposed in Chapter 3, that does not require signal extension. The algorithm is similar in spirit to the one developed for 2-D signals, with increased resources allocated to finding the coarsest resolution subband coefficients of the 3-D wavelet decomposition. The algorithm is then used in two gradient based video editing tasks: one in which content of a video is replaced with content from another, and one in which two videos are combined with the purpose of simulating a transparency effect. Results illustrate the good performance of the proposed technique in reconstructing video sequences from gradient measurements and indicate that the proposed algorithm is suitable for gradient based video processing applications. The advantage of the developed technique is most evident when the signal of interest has significant differences between its dimensions, typically the case of video signals, where the time dimension is typically larger than the two spatial dimensions.

The findings of Chapters 2 and 3, and in addition the powerful potential of light field scene representations, motivate expanding the previously described framework with the purpose of developing an algorithm that can be used to reconstruct a light

field from a given gradient dataset. Current sensor capabilities impose what is known as a spatio-angular resolution trade-off on the photographs recorded with the help of light field cameras (i.e. camera outputs a relatively small number of relatively high resolution views of the same scene). This makes the idea of obtaining an intermediate signal with a symmetric region of support along all dimensions impractical, as a very large amount of unnecessary data would be generated. Instead, a non-symmetric 4-D wavelet decomposition is obtained from the light field gradient data. Several ways to approximate the coarsest resolution subband coefficients of this 4-D wavelet decomposition are proposed, based on the similarities between adjacent views in a light field. The visual impact of approximating these coefficients on the quality of the reconstructed light field is studied on several datasets.

In Chapter 5, ideas inspired by successful gradient based image and video processing algorithms are explored in the context of four dimensional signal processing. In particular, a new gradient based technique for light field fusion is proposed and two light field editing applications are studied. In these applications, a 4-D gradient data set is first generated by combining information from the gradient of several source light fields. Then, a light field is recovered from this artificially generated gradient by using the reconstruction technique developed in Chapter 5. The results included show that gradient domain processing remains an attractive alternative despite the high dimensionality of the data.

6.2 Future Directions

6.2.1 Vision correction

The need for enhanced vision quality surrounds us, from people with temporary or permanent eyesight impairments, to robots operating in harsh visibility conditions, or telescopes imaging far in space. Developing new and efficient algorithms to improve the quality of the images perceived by either humans or machines is thus worthwhile exploring in future work.

In the past decades, novel ideas [90], [91], [92], [65] propose using both the displays and the rendered images to compensate for a person's specific vision problems. Such displays are called computational displays and eliminate the need for wearable visual aids. These advances are based on notions from light field theory and we believe that the developments of this thesis are a valuable tool that can be utilized to further

progress in this direction.

6.2.2 Improved facial recognition

The need for algorithms that are able to correctly identify a person arises in a wide range of places that share a requirement for a high degree of security, such as school grounds, pediatric wards in hospitals, financial institutions, museums, airports or border crossings. The problem of face recognition is not new, and designing algorithms to successfully solve this problem and identify a person from has gathered research interest for many years now [93], [94], [95].

Light field cameras offer tremendous advantages over conventional cameras, from the point of view of the richness of information that they produce from just a single shutter actuation. In a way, a light field camera outputs not a single image of a face pose, but rather, an entire dataset consisting of slightly different poses of the same face. This wealth of information should make face recognition algorithms more robust to pose variations, a quality algorithms often strive to achieve. In addition, due to the proven potential of light field cameras to eliminate glass reflections [76], [77], [78], challenging problems such as identifying person through closed tinted car windows can be tackled. These ideas are currently explored by border patrol entities [96] and the developments of Chapter 4 and 5 of this thesis have potential in this direction.

6.2.3 Light field video editing

Light field video acquisition in itself is in its early stages, with pioneering work using the Lytro cameras done in [97]. Sabater et al. [98] devise a pipeline to acquire multi view light-field videos. They also make a very important point: that both the acquisition and the processing of light field videos currently largely depend on the application at hand. In reality though, acquisition systems *are* different and so, in order to truly benefit from this new way of representing light in a scene, exploring the possibility to develop a framework to successfully interpret and combine information from differently acquired light fields seems like the next natural step to take.

Bibliography

[1] Gordon Wetzstein. Synthetic light field archive. http://web.media.mit.edu/~gordonw/SyntheticLightFields/index.php. Accessed: 2018-01-30.

[2] Adrian Jarabo, Belen Masia, Adrien Bousseau, Fabio Pellacini, and Diego Gutierrez. How do people edit light fields? *ACM Transactions on Graphics*, 33(4):146:1–146:10, July 2014.

[3] University of Konstanz and the HCI at Heidelberg University. 4D light field dataset. http://hci-lightfield.iwr.uni-heidelberg.de/, 2018. Accessed: 2018-01-10.

[4] Stanford University Computer Graphics Laboratory. The (new) Stanford light field archive. http://lightfield.stanford.edu/lfs.html. Accessed: 2018-01-30.

[5] Sidd Bikkannavar and David Redding. The end of the blur. *IEEE Spectrum*, 47(3):46–53, March 2010.

[6] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics*, 21(3):249–256, July 2002.

[7] Patrick Perez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *Proceedings of SIGGRAPH*, pages 313–318, Jul. 2003.

[8] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *Eighth European Conference on Computer Vision*, volume 4, pages 377–389, 2004.

[9] Hongcheng Wang, Ning Xu, Ramesh Raskar, and Narendra Ahuja. Videoshop: A new framework for spatio-temporal video editing in gradient domain. *Graph. Models*, 69(1):57–70, January 2007.

[10] Ioana S. Sevcenco and Panajotis Agathoklis. Video editing in the gradient domain using a wavelet based 3-D reconstruction algorithm and an iterative Poisson solver. In *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 205–209, Aug 2015.

[11] Robert T. Frankot and Rama Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:439–451, Jul. 1988.

[12] Amit Agrawal, Rama Chellappa, and Ramesh Raskar. An algebraic approach to surface reconstruction from gradient fields. In *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 174–181. IEEE, 2005.

[13] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[14] Anthony J. Roberts. Simple and fast multigrid solution of Poisson's equation using diagonally oriented grids. *ANZIAM J.*, 43(E):E1–E36, July 2001. [Online] http://anziamj.austms.org.au/V43/E025 [2 Jul 2001].

[15] Peter John Hampton, Panajotis Agathoklis, and Colin Bradley. A new wave-front reconstruction method for adaptive optics system using wavelets. *IEEE Journal of Selected Topics in Signal Processing*, 2(5):781–792, Oct. 2008.

[16] Ioana S Sevcenco, Peter J Hampton, and Panajotis Agathoklis. A wavelet based method for image reconstruction from gradient data with applications. *Multidimensional Syst. Signal Process.*, 26(3):717–737, July 2015.

[17] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[18] Sujoy Paul, Ioana S. Sevcenco, and Panajotis Agathoklis. Multi-exposure and multi-focus image fusion in gradient domain. *Journal of Circuits, Systems and Computers*, 25(10), 2016.

[19] Mohammad Bagher Akbari Haghighat, Ali Aghagolzadeh, and Hadi Seyedarabi. Multi-focus image fusion for visual sensor networks in dct domain. *Computers and Electrical Engineering*, 37(5):789–797, 2011.

[20] Y. Song, M. Li, Q. Li, and L. Sun. A new wavelet based multi-focus image fusion scheme and its application on optical microscopy. In *2006 IEEE International Conference on Robotics and Biomimetics*, pages 401–405, Dec 2006.

[21] Xuejun Li and Minghui Wang. Research of multi-focus image fusion algorithm based on gabor filter bank. In *2014 12th International Conference on Signal Processing (ICSP)*, pages 693–697, Oct 2014.

[22] Biswajit Biswas, Ritamshirsa Choudhuri, Kashi Nath Dey, and Amlan Chakrabarti. A new multi-focus image fusion method using principal component analysis in shearlet domain. In *Proceedings of the 2Nd International Conference on Perception and Machine Intelligence*, PerMIn '15, pages 92–98, New York, NY, USA, 2015. ACM.

[23] Rishu Garg, Preeti Gupta, and Harvinder Kaur. Survey on multi-focus image fusion algorithms. In *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, pages 1–5, March 2014.

[24] Wang Yajie, Ye Yongsheng, Ran Xiaoyan, Wu Yanyan, and Shi Xiangbin. A multi-focus color image fusion method based on edge detection. In *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pages 4294–4299, May 2015.

[25] Yibo Chen and Wai-Kuen Cham. Edge model based fusion of multi-focus images using matting method. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1840–1844, Sept 2015.

[26] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *Computer Graphics and Applications, 2007. PG '07. 15th Pacific Conference on*, pages 382–390, Oct 2007.

[27] Amina Saleem, Azeddine Beghdadi, and Boualem Boashash. Image fusion-based contrast enhancement. *EURASIP Journal on Image and Video Processing*, 2012(1):10, May 2012.

[28] Rui Shen, Irene Cheng, Jianbo Shi, and Anup Basu. Generalized random walks for fusion of multi-exposure images. *IEEE Transactions on Image Processing*, 20(12):3634–3646, Dec 2011.

[29] Shutao Li and Xuodong Kang. Fast multi-exposure image fusion with median filter and recursive filter. *IEEE Transactions on Consumer Electronics*, 58(2):626–632, May 2012.

[30] Jun Kong, Ruajan Wang, Yingha Lu, Xue Feng, and Jingbuo Zhang. A novel fusion approach of multi-exposure image. In *EUROCON 2007 - The International Conference on "Computer as a Tool"*, pages 163–169, Sept 2007.

[31] D. A. Socolinsky and L. B. Wolff. Multispectral image visualization through first-order fusion. *IEEE Transactions on Image Processing*, 11(8):923–931, Aug 2002.

[32] Chao Wang, Qiong Yang, Xiaoou Tang, and Zhongfu Ye. Salience preserving image fusion with dynamic range compression. In *2006 International Conference on Image Processing*, pages 989–992, Oct 2006.

[33] Kenji Hara, Kohei Inoue, and Kiichi Urahama. A differentiable approximation approach to contrast-aware image fusion. *IEEE Signal Processing Letters*, 21(6):742–745, June 2014.

[34] Kede Ma and Zhou Wang. Multi-exposure image fusion: A patch-wise approach. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1717–1721, Sept 2015.

[35] Karel Zuiderveld. Graphics gems iv. chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

[36] V P S Naidu. Image fusion technique using multi-resolution singular value decomposition. *Defence Science Journal*, 61(5):479–484, Sep. 2011.

[37] Sujoy Paul, Ioana Sevcenco, and Pan Agathoklis. Multi-exposure and Multi-focus Image Fusion in Gradient Domain. https://bit.ly/2xaSb4j. Accessed: 2018-05-21.

[38] HDR images from the CAVE (Columbia Automated Vision Environment) Lab-Source exposures courtesy of S. Nayar. https://www.cs.huji.ac.il/~danix/hdr/pages/columbia.html. Accessed: 2016-05-21.

[39] Jun Hu, Orazio Gallo, Kari Pulli, and Xiaobai Sun. HDR deghosting: How to deal with saturation? In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1163–1170, June 2013.

[40] Tom Mertens. Data and software implementation of exposure fusion algorithm. http://www.mericam.net/. Accessed: May 21, 2016.

[41] Mohammad Haghighat. Multi-focus image fusion in dct domain. https://bit.ly/2s3pr8U. Accessed: May 21, 2018.

[42] J. van de Weijer. Image data sets.

http://lear.inrialpes.fr/people/vandeweijer/.

[43] Salvador Gabarda and Gabriel Cristóbal. Blind image quality assessment through anisotropy. *J. Opt. Soc. Am. A*, 24(12):B42–B51, Dec 2007.

[44] Stanford University Computer Graphics Laboratory. The digital Michelangelo project. https://graphics.stanford.edu/projects/mich/. Accessed:2018-06-14.

[45] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[46] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13, July 2013.

[47] D. Sun, M. E. Rettmann, D. R. Holmes Iii, C. Linte, B. Cameron, J. Liu, D. Packer, and R. A. Robb. Anatomic surface reconstruction from sampled point cloud data and prior models. *Stud Health Technol Inform*, 196:387–393, 2014.

[48] W. Liu, Y. Cheung, A. Sawant, and D. Ruan. A robust real-time surface reconstruction method on point clouds captured from a 3D surface photogrammetry system. *Med Phys*, 43(5):2353, May 2016.

[49] Florian Bernard, Luis Salamanca, Johan Thunberg, Alexander Tack, Dennis Jentsch, Hans Lamecker, Stefan Zachow, Frank Hertel, Jorge M. Gonçalves, and Peter Gemmar. Shape-aware surface reconstruction from sparse data. *CoRR*, abs/1602.08425, 2016.

[50] Nolan Goodnight, Cliff Woolley, Gregory Lewin, David Luebke, and Greg Humphreys. A multigrid solver for boundary value problems using programmable graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, HWWS '03, pages 102–111, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[51] James McCann and Nancy S. Pollard. Real-time gradient-domain painting. *ACM Trans. Graph.*, 27(3):93:1–93:7, August 2008.

[52] Matthew Harker and Paul O'leary. Regularized reconstruction of a surface from its measured gradient field. *J. Math. Imaging Vis.*, 51(1):46–70, January 2015.

[53] Jun Zhang. Fast and high accuracy multigrid solution of the three dimensional Poisson equation. *J. Comput. Phys.*, 143(2):449–461, July 1998.

[54] Peter John Hampton and Panajotis Agathoklis. Comparison of Haar wavelet-based and Poisson-based numerical integration techniques. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 1623–1626, 2010.

[55] VIPs Lab. Database: Images and video clips (2). http://see.xidian.edu.cn/vipsl/database_Video.html. Accessed: 2018-06-17.

[56] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Stanford University Computer Science Tech Report CSTR 2005-02*, April 2005.

[57] Bennett Wilburn. *High Performance Imaging Using Arrays of Inexpensive Cameras*. PhD thesis, Stanford University, 2005.

[58] Jason C. Yang, Matthew Everett, Chris Buehler, and Leonard McMillan. A real-time distributed light field camera. In *Proceedings of the 13th Eurographics Workshop on Rendering*, EGRW '02, pages 77–86, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[59] Lytro. https://support.lytro.com/hc/en-us. Accessed: June, 2018.

[60] Raytrix. https://raytrix.de/. Accessed: June, 2018.

[61] V Vaish, M Levoy, R Szeliski, C. L. Zitnick, and Sing Bing Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2331–2338, 2006.

[62] Donald G Dansereau. *Plenoptic signal processing for robust vision in field robotics*. PhD thesis, The University of Sydney, 2014.

[63] Matthew Hirsch, Gordon Wetzstein, and Ramesh Raskar. A compressive light field projection system. In *ACM SIGGRAPH 2014 Emerging Technologies*, SIGGRAPH '14, pages 2:1–2:1, New York, NY, USA, 2014. ACM.

[64] Andrew Maimone, Gordon Wetzstein, Matthew Hirsch, Douglas Lanman, Ramesh Raskar, and Henry Fuchs. Focus 3d: Compressive accommodation display. *ACM Trans. Graph.*, 32(5):153:1–153:13, October 2013.

[65] Fu-Chung Huang, Gordon Wetzstein, Brian A. Barsky, and Ramesh Raskar. Eyeglasses-free display: Towards correcting visual aberrations with computational light field displays. *ACM Trans. Graph.*, 33(4):59:1–59:12, July 2014.

[66] Marta Ortin, Adrian Jarabo, Belen Masia, and Diego Gutierrez. Analyzing interfaces and workflows for light field editing. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1162–1172, Oct 2017.

[67] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4D light fields. In *Computer Vision – ACCV 2016*, pages 19–34, Cham, 2017. Springer International Publishing.

[68] Qiang Fu, Zhiliang Zhou, Yan Yuan, and Bin Xiangli. Image quality evaluation of light field photography. 7867, 01 2011.

[69] Aaron Isaksen, Leonard McMillan, and Steven J. Gortler. Dynamically reparameterized light fields. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 297–306, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[70] Marc Levoy, Billy Chen, Vaibhav Vaish, Mark Horowitz, Ian McDowall, and Mark Bolas. Synthetic aperture confocal imaging. *ACM Trans. Graph.*, 23(3):825–834, August 2004.

[71] Daniel Reiter Horn and Billy Chen. Lightshop: Interactive light field manipulation and rendering. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, I3D '07, pages 121–128, New York, NY, USA, 2007. ACM.

[72] Akira Kubota, Kiyoharu Aizawa, and Tsuhan Chen. Reconstructing dense light field from array of multifocus images for novel view synthesis. *IEEE Transactions on Image Processing*, 16(1):269–279, Jan 2007.

[73] Ralph Gross, Iain Matthews, and Simon Baker. Eigen light-fields and face recognition across pose. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–7, May 2002.

[74] Ralph Gross, Iain Matthews, and Simon Baker. Appearance-based face recognition and light-fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):449–465, April 2004.

[75] Raghavendra R., Bian Yang, Kiran B Raja, and C Busch. A new perspective – face recognition with light-field camera. In *2013 International Conference on Biometrics (ICB)*, pages 1–8, June 2013.

[76] Qiaosong Wang, Haiting Lin, Yi Ma, Sing Bing Kang, and Jingyi Yu. Automatic layer separation using light field imaging. *CoRR*, abs/1506.04721, 2015.

[77] Ole Johannsen, Antonin Sulc, and Bastian Goldluecke. What sparse light field coding reveals about scene structure. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3262–3270, June 2016.

[78] Tingtian Li and Daniel P K Lun. A novel reflection removal algorithm using the light field camera. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, May 2018.

[79] M. Umair Mukati and Bahadir K. Gunturk. Light field stitching for extended synthetic aperture. *CoRR*, abs/1611.05003, 2016.

[80] Clemens Birklbauer and Oliver Bimber. Panorama light-field imaging. In *ACM SIGGRAPH 2012 Posters*, SIGGRAPH '12, pages 61:1–61:1, New York, NY, USA, 2012. ACM.

[81] Clemens Birklbauer and Oliver Bimber. Panorama light-field imaging. *Comput. Graph. Forum*, 33(2):43–52, May 2014.

[82] X. Guo, Z. Yu, S. Kang, H. Lin, and J. Yu. Enhancing light fields through ray-space stitching. *Visualization and Computer Graphics, IEEE Transactions on*, PP(99):1–1, 2015.

[83] Pratul P. Srinivasan, Ren Ng, and Ravi Ramamoorthi. Light field blind motion deblurring. *CoRR*, abs/1704.05416, 2017.

[84] Farnoud Kazemzadeh, Chao Jin, Sara Molladavoodi, Yu Mei, Monica B. Emelko, Maud B. Gorbet, and Alexander Wong. Lens-free spectral light-field fusion microscopy for contrast- and resolution-enhanced imaging of biological specimens. *Opt. Lett.*, 40(16):3862–3865, Aug 2015.

[85] Fang-Lue Zhang, Jue Wang, Eli Shechtman, Zi-Ye Zhou, Jia-Xin Shi, and Shi-Min Hu. Plenopatch: Patch-based plenoptic image manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 23(5):1561–1573, May 2017.

[86] Alexandre Vieira, Helder Duarte, Cristian Perra, Luis Tavora, and Pedro Assuncao. Data formats for high efficiency coding of lytro-illum light fields. In *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 494–497, Nov 2015.

[87] Irene Viola, Martin Řeřábek, and Touradj Ebrahimi. A new approach to subjectively assess quality of plenoptic content. volume 9971, pages 9971 – 9971 – 13, 2016.

[88] Irene Viola, Martin Řeřábek, and Touradj Ebrahimi. Comparison and evaluation of light field image coding approaches. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1092–1106, Oct 2017.

[89] J. Tumblin, A. Agarwal, and A. Raskar. Why I want a gradient camera. In *Proceeding of IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, volume 1, pages 103–110, 2005.

[90] M. Alonso and Armando Barreto. Pre-compensation for high-order aberrations of the human eye using on-screen image deconvolution. In *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, volume 1, pages 556–559 Vol.1, Sept 2003.

[91] Fu-Chung Huang, Douglas Lanman, Brian A. Barsky, and Ramesh Raskar. Correcting for optical aberrations using multilayer displays. *ACM Trans. Graph.*, 31(6):185:1–185:12, November 2012.

[92] Vitor F. Pamplona, Manuel M. Oliveira, Daniel G. Aliaga, and Ramesh Raskar. Tailored displays to compensate for visual aberrations. *ACM Trans. Graph.*, 31(4):81:1–81:12, July 2012.

[93] Matthew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, Jun 1991.

[94] Jian Yang, D. Zhang, A. F. Frangi, and Jing-yu Yang. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, Jan 2004.

[95] Jun-Cheng Chen, Vishal M. Patel, and Rama Chellappa. Unconstrained face verification using deep CNN features. *CoRR*, abs/1508.01722, 2015.

[96] Russell Brandom. Feds look to experimental light-field cameras to ID drivers at the border. https://bit.ly/2rszapm. Accessed: 2018-06-16.

[97] Chamira U. S. Edussooriya, Len T. Bruton, and Panajotis Agathoklis. Enhancing moving objects in light field videos using 5-d IIR adaptive depth-velocity filters. In *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 169–173, Aug 2015.

[98] Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, and Valerie Alli. Dataset and pipeline for multi-view light-field video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1743–1753, July 2017.

[99] D. S. Watkins. *Fundamentals of Matrix Computations*. John Wiley & Sons, Inc., New York, USA, second edition, 2002.

[100] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[101] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[102] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.

[103] Lixin Shi, Haitham Hassanieh, Abe Davis, Dina Katabi, and Fredo Durand. Light field reconstruction using sparsity in the continuous fourier domain. *ACM Trans. Graph.*, 34(1):12:1–12:13, December 2014.

[104] Ioana S. Sevcenco and Panajotis Agathoklis. The role of the coarsest resolution subband in the wavelet-based reconstruction of signals from gradients. *Journal of Physics: Conference Series*, 933(1), 2018.

[105] Ramesh Raskar, Adrian Ilie, and Jingyi Yu. Image fusion for context enhancement and video surrealism. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering*, NPAR '04, pages 85–152, New York, NY, USA, 2004. ACM.

[106] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski. Video matting of complex scenes. *ACM Transactions on Graphics*, 21(3):243–248, July 2002.

[107] R. W. Hockney. A fast direct solution of poisson's equation using fourier analysis. *J. ACM*, 12(1):95–113, January 1965.

[108] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4D light fields. In *Asian Conference on Computer Vision*. Springer, 2016.

[109] Richard Socher. Free open source png to eps and jpg to eps converter for windows. https://bit.ly/Rbtgno. Accessed: 2018-04-10.

[110] MATLAB. *version 9.4 (R2018a)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[111] Donald Dansereau. Light-field toolbox for Matlab v0.4. https://bit.ly/2JptsOT.

[112] Bian Ruping, Li Jing, Yang Tao, Lu Zhaoyang, Li Zhongzhen, and Zhang Fangbing. Removing reflection from images based on bp neural network of motion field. In *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pages 768–771, Nov 2016.

[113] Belen Masia. Computational imaging and displays: Capturing and displaying richer representations of the world. *IEEE Computer Graphics and Applications*, 38(2):112–120, Mar 2018.

[114] Gaochang Wu, Belen Masia, Adrian Jarabo, Yuchen Zhang, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu. Light field image processing: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):926–954, Oct 2017.

[115] Francisco Calderon, Carlos Parra, and Cesar Nino. A modular robotic platform for visual perception and computational photography. In *2016 IEEE Colombian Conference on Robotics and Automation (CCRA)*, pages 1–6, Sept 2016.

[116] Ole Johannsen, Antonin Sulc, and Bastian Goldluecke. On linear structure from motion for light field cameras. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 720–728, Dec 2015.

[117] Zhunping Zhang, Lifeng Wang, Baining Guo, and Heung-Yeung Shum. Feature-based light field morphing. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 457–464, New York, NY, USA, 2002. ACM.

Appendix A

Example for computing coarsest resolution coefficients 2-D case

A small size example relevant to the procedure described in Chapter 2 for finding the coarsest resolution array $\hat{\Phi}^0_{LL}$ in the 2-D case is included here.

Let $\Phi \in \mathbb{R}^{2^M \times 2^N}$, with $M = 2$ and $N = 3$. The maximum number of levels in the wavelet decomposition of Φ is 2. The size of $\hat{\Phi}^0_{LL}$ is 1×2 and the values of its elements can be found from the signal Φ as illustrated in the diagram shown in Figure A.1.

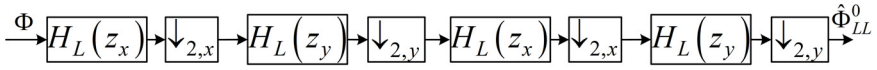
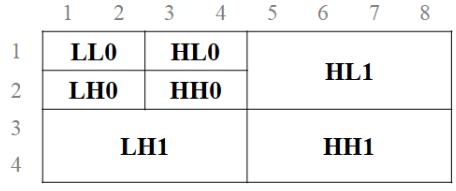


Figure A.1: Obtaining the coarsest resolution array $\hat{\Phi}^0_{LL}$ of a 4×8 signal from the signal

	1	2	3	4	5	6	7	8
1	**3**	**8**	**5**	**7**	**7**	**10**	**5**	**16**
2	**14**	**10**	**3**	**2**	**5**	**4**	**2**	**3**
3	**3**	**13**	**1**	**14**	**10**	**4**	**2**	**0**
4	**15**	**8**	**12**	**4**	**9**	**4**	**16**	**3**

(a) Example of a 2-D signal Φ.



(b) Wavelet decomposition $\hat{\Phi}$.

Figure A.2: Numerical 2-D example for visualizing the coarsest resolution array. Signal Φ versus wavelet decomposition $\hat{\Phi}$

The two elements of $\hat{\Phi}^0_{LL}$ (marked LL_0 in Figure A.2b are the sums of all values in the two 4×4 square arrays that make up the 2-D signal Φ, scaled by $1/4$. In the general case, when $M < N$, the elements of $\hat{\Phi}^0_{LL}$ are sums of the all values in the 2^{N-M} square arrays with size $2^M \times 2^M$ that make up Φ, scaled by $1/2^M$.

The question that the algorithm presented in Section 2.5.2 of Chapter 2 of this dissertation is trying to address is how to find the two (or, more generally, the 2^{N-M}) elements of $\hat{\Phi}^0_{LL}$ from the signal gradient.

	1	2	3	4	5	6	7
1	**5**	**-3**	**2**	**0**	**3**	**-5**	**11**
2	**-4**	**-7**	**-1**	**3**	**-1**	**-2**	**1**
3	**10**	**-12**	**13**	**-4**	**-6**	**-2**	**-2**
4	**-7**	**4**	**-8**	**5**	**-5**	**12**	**-13**

(a) Φ_x

	1	2	3	4	5	6	7	8
1	**11**	**2**	**-2**	**-5**	**-2**	**-6**	**-3**	**-13**
2	**-11**	**3**	**-2**	**12**	**5**	**0**	**0**	**-3**
3	**12**	**-5**	**11**	**-10**	**-1**	**0**	**14**	**3**

(b) Φ_y

Figure A.3: Gradient of numerical 2-D example.

In Figure A.3, the two directional components of the gradient of Φ are shown. The elements of Φ_x and Φ_y are differences between neighbouring values of Φ in the horizontal and vertical directions, respectively.

The objective thus is to find $\hat{\Phi}^0_{LL}$, from Φ_x and/ or Φ_y.

Let the two elements of $\hat{\Phi}^0_{LL}$ be denoted by λ_1 and λ_2. These elements are given by:

$$\lambda_1 = \frac{1}{4}\sum_{y,x=1}^{4} \Phi\left(y,x\right) = 30.5 \tag{A.1}$$

$$\lambda_2 = \frac{1}{4}\sum_{\substack{y\in\{1,\ldots,4\}\\ x\in\{5,\ldots,8\}}} \Phi\left(y,x\right) = 25 \tag{A.2}$$

Consider the following linear combination of column vectors of Φ_x:

$$c_1 + 2c_2 + c_3 = \begin{bmatrix} -\left[\Phi\left(1,1\right)+\Phi\left(1,2\right)\right]+\Phi\left(1,3\right)+\Phi\left(1,4\right) \\ -\left[\Phi\left(2,1\right)+\Phi\left(2,2\right)\right]+\Phi\left(2,3\right)+\Phi\left(2,4\right) \\ -\left[\Phi\left(3,1\right)+\Phi\left(3,2\right)\right]+\Phi\left(3,3\right)+\Phi\left(3,4\right) \\ -\left[\Phi\left(4,1\right)+\Phi\left(4,2\right)\right]+\Phi\left(4,3\right)+\Phi\left(4,4\right) \end{bmatrix} \tag{A.3}$$

where c_1, c_2 and c_3 are the first three columns of Φ_x.

Adding up the elements of the column vector on the right hand-side of Eq. A.3, we can compute:

$$u_1 = -\sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{1,2\}}} \Phi(y,x) + \sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{3,4\}}} \Phi(y,x) \tag{A.4}$$

Similarly, from c_3, c_4, c_5 and c_5, c_6, c_7, we can compute:

$$u_2 = -\sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{3,4\}}} \Phi(y,x) + \sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{5,6\}}} \Phi(y,x) \tag{A.5}$$

$$u_3 = -\sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{5,6\}}} \Phi(y,x) + \sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{7,8\}}} \Phi(y,x) \tag{A.6}$$

For simplicity, we denote:

$$s_1 = \sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{1,2\}}} \Phi(y,x) \tag{A.7}$$

$$s_2 = \sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{3,4\}}} \Phi(y,x) \tag{A.8}$$

$$s_3 = \sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{5,6\}}} \Phi(y,x) \tag{A.9}$$

$$s_4 = \sum_{\substack{y\in\{1,\dots,4\}\\ x\in\{7,8\}}} \Phi(y,x) \tag{A.10}$$

Substituting the summations in Equations A.4–A.6 with the shorter notation introduced in Equations A.7–A.10, we can write:

$$-s_1 + s_2 = u_1 \tag{A.11}$$

$$-s_2 + s_3 = u_2 \tag{A.12}$$

$$-s_3 + s_4 = u_3 \tag{A.13}$$

In the system of equations A.11–A.13, u_1, u_2 and u_3 are scalars, obtained from the x gradient component of Φ, namely Φ_x. Recall that our interest is to find λ_1 and λ_2, which in the newly introduced notation are: $\lambda_1 = s_1 + s_2$ and $\lambda_2 = s_3 + s_4$. The approach to find these summations is to solve the system of equations A.11– A.13,

find s_1, s_2, s_3 and s_4, and then add them up pairwise to get the desired values λ_1 and λ_2, i.e. the elements of $\hat{\Phi}^0_{LL}$.

The system described by the equations A.11–A.13 has three equations and four unknowns, and is thus under-determined. In order to solve it exactly, **we assume known the average value of the 2-D signal**.

Let the average value of the signal Φ be denoted by m. This yields the following equation:

$$s_1 + s_2 + s_3 + s_4 = 2^{M+N} \cdot m \tag{A.14}$$

We now rewrite the system of equations A.11–A.14 in matrix form:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} 2^{M+N} \cdot m \\ u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{A.15}$$

Denoting $A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$, $\lambda = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$ and $v = \begin{bmatrix} 2^{M+N} \cdot m \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}$, we can rewrite Eq. A.15 as:

$$A\lambda = v \tag{A.16}$$

where A is fixed and known, v is also known with three values obtained from the Φ_x derivative of Φ and the last one from the mean value of Φ, and λ is the unknown.

Solving Eq. A.16 for λ yields:

$$\lambda = A^{-1}v \tag{A.17}$$

Recall that $\lambda = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \end{bmatrix}^T$. With λ found, we can now compute the desired values $\lambda_1 = s_1 + s_2$ and $\lambda_2 = s_3 + s_4$.

In the numerical example illustrated by Figures A.2a and A.3a, we have:

$$v = \begin{bmatrix} 222 \\ -26 \\ 5 \\ -6 \end{bmatrix} \tag{A.18}$$

Thus:

$$\lambda = \frac{1}{2^M} A^{-1} v = \begin{bmatrix} 18.5 \\ 12 \\ 13.25 \\ 11.75 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \tag{A.19}$$

As such, $\lambda_1 = s_1 + s_2 = 30.5$ and $\lambda_2 = s_3 + s_4 = 25$, which are the same as the values obtained directly from the signal, as can be seen from Eq. A.1 and Eq. A.2.

Appendix B

Two dimensional Poisson solver

Here is a brief description of the Poisson solver used in chapter 2. This is a Jacobi iterative Poisson solver, and Hampton et al. [54] highlight its benefits when applied at each resolution level of an image, during the synthesis step.

In the notation of this thesis, the Poisson solver is given by:

$$\Phi(k+1) = \Phi(k) + 0.25\left[\Delta\Phi(k) - \left(\frac{\partial\tilde{\Phi}_x}{\partial x} + \frac{\partial\tilde{\Phi}_y}{\partial y}\right)\right] \tag{B.1}$$

where:

- $\Phi(k)$ is the reconstructed signal at a given resolution level L
- $\Delta\Phi(k) = \frac{\partial^2\Phi(k)}{\partial x^2} + \frac{\partial^2\Phi(k)}{\partial y^2}$ is the Laplacian of $\Phi(k)$
- $\tilde{\Phi}_x$ and $\tilde{\Phi}_y$ are subsampled lowpass filtered versions of the given gradient obtained and stored in the analysis step.

This solver extends naturally to 3-D and 4-D signals, and is used where referred to in Chapters 3, 4 and 5, respectively.

The role of the Poisson solver in the applications included in this thesis is to smooth out the errors caused by the non-integrable nature of the gradient data, from which the signal has to be reconstructed. This is representative for situations when the gradient data is generated from multiple signals, to satisfy certain requirements in editing applications.