Aalen University

Zentrum für optische Technologien- ZOT

Optische Messtechnik - OMT

Fast Numerical Scalar Diffraction Simulation for Optical Neural Networks

A Master Thesis

In Partial Fulfillment for the Degree of

Master of Science

(M.Sc.)

By Christian Eder, B.Eng.

52944

January 2021

1st Auditor: Prof. Dr. Andreas Heinrich

2nd Auditor: Prof. Dr. Rainer Börret

I Abstract

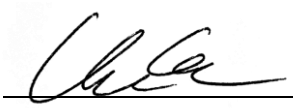
Calculation of optical free-space propagation for diffractive optical neural network is a matter that lacks computational efficiency, due to the novelty of the field. To enable fast ex situ training for diffractive neural networks operating at visible and near-infrared wavelengths, a bandwidth limited variation of the angular spectrum method is applied to the free-space propagation problem. The computational efficiency is achieved by optimizing the sampling conditions of the calculated fields and therefore decreasing the necessary field size. An in-depth view onto the derivation of the angular spectrum method from the Rayleigh-Sommerfeld integrals is given and the derived sampling conditions examined. Additionally, a theoretical concept of a convolutional diffractive neural network layer is proposed. By applying the in this work derived methods, the training time of diffractive neural networks can be decreased dramatically and makes the use of a wider range of optical wavelengths possible.

Keywords: Neural Networks, Deep Learning, Diffraction Simulation, Photonic Networks, Rayleigh-Sommerfeld, Angular Spectrum, Holography, Optical Computing

II Statutory Declaration

I hereby declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material that has been quoted either literally or by content from the used sources.

Aalen, 19th of January 2021



Christian Eder

III Table of Contents

1 Introduction 6

2 Artificial Neural Networks 9

2.1 Basic Concept of Artificial Neural Networks 9

2.1.1 Neurons in an NN 9

2.1.2 Shallow Artificial Neural Networks 11

2.1.3 Deep Neural Networks 14

2.2 Training of a Basic Neural Network 15

2.2.1 The Loss Function 15

2.2.2 Gradient Descent 17

2.2.3 Backpropagation Model 20

2.3 Activation Functions and Nonlinearity 24

2.4 Convolutional Neural Networks 30

3 Optical Neural Networks 36

3.1 State of the Art 36

3.2 Deep Diffractive Neural Networks 42

3.2.1 Mathematical Model of an Optical Feed-Forward Network 42

3.2.2 Nonlinearity in D^2NNs 45

3.2.3 Diffractive vs. Classical Neural Networks 46

3.2.4 Backpropagation in Deep Diffractive Neural Networks 49

3.2.5 Convolutional Deep Diffractive Neural Networks 51

3.3 Hyphotheses & Further Developments of F-/D^2NNs 52

4 Numerical Scalar Diffraction Simulation 57

4.1 Analytical description of scalar diffraction 57

4.1.1 From the Maxwell Equations to the Scalar Wave Equation 58

4.1.2 The Helmholtz Equation 60

4.1.3 Fresnel-Kirchhoff's Formulation of Diffraction 62

4.1.4 The Rayleigh-Sommerfeld Diffraction Formulation 66

4.1.5 Linear Systems & Transfer Functions 68

4.2 The Angular Spectrum Method 71

4.2.1 The Discrete Fourier Transformation of Sampled Fields 73

4.2.2 Zero-Padding for Discrete Linear Convolutions 75

4.2.3 Band-limited Angular Spectrum Method (BLAS) 79

4.3 Complex Amplitude Wavefront Modulation 85
4.4 BLAS algorithm for complex wave propagation in MATLAB 89
4.4.1 Description of the Propagation Algorithm 89
4.4.2 Calculating the FSP Transfer Function 91
4.5 Standalone Scalar Diffraction Simulation 94
4.5.1 Definition of the Input Data 94
4.5.2 Description of the Standalone Simulation Algorithm 96

5 Experimental 98
5.1 Computational Speed of the Bandlimited Angular Spectrum Method 98
5.1.1 Experimental Setup 99
5.1.2 Results 102
5.1.3 Discussion 104
5.2 Scalar Diffraction Simulation of a Convolution Unit 108
5.2.1 Experimental Setup Part 1 110
5.2.2 Results Part 1 112
5.2.3 Discussion Part 1 114
5.2.4 Experimental Setup Part 2 114
5.2.5 Results Part 2 115
5.2.6 Discussion Part 2 117
5.2.7 Experimental Setup Part 3 118
5.2.8 Results Part 3 121
5.2.9 Discussion Part 3 126
5.2.10 Experimental Setup Part 4 129
5.2.11 Results Part 4 133
5.2.12 Discussion Part 4 135
5.3 Scalar Diffraction Simulation with Real Data 136
5.3.1 Setup 137
5.3.2 Results 139
5.3.3 Discussion 143
5.4 Scalar Diffraction Simulation of Holographic Surfaces 143
5.4.1 Experimental Setup 144
5.4.2 Results 149
5.4.3 Discussion 151

6 Summary 153

7 Outlook 157

8 Bibliography 159

9 Table of Figures 168

1 Introduction

Since the invention of digital computers in the last century, computing power has roughly doubled every two years over the last 50 years, known as Moore's law [1]. This was possible due to the increasing transistor density on processor chips. Up until now, technological advances keep up with the demands of decreasing component sizes, although since 2010 the development has slowed down and the end of validity of Moore's law has been predicted to be in the next decades [2, 3]. Alternative methods for digital or even analog computing have been researched, but none of them reached the computational efficiency of classical transistor computing. One promising alternative is quantum computing which is already usable as of today [4], but has a very specific application range.

Another promising alternative are optical processors[1], which use light as information medium. Research on this topic is almost as old as the electronical computing itself. Optical computing arguably never reached the state of matureness of electronic processors but show promising advantages in parallelism and information processing speeds. [5]

A technology benefiting from high degree of parallel computing is neural computing. In recent years the interest in approaches for computing algorithms has shifted towards neural architectures as the availability of large data increased dramatically [6]. The field of "deep learning" is out for creating artificial intelligence and therefore creating a new kind of information processing which resamples the way the human brain processes information rather than digital computers do. Sparked by the increasing interest in neural computing, optical information processing and optical neural computing has seen an increase in scientific publications as well. A analytical search for the number of publications per year on the *web of science* [7] shows a shared increase in the last decade for the keywords: "optical neural network" and "deep learning". See Figure 1-1 for reference. But the total number of optical related publications is still a new field which has not found one common base for further research, as still many fundamentally different concepts are explored [8].

Based on the concept of optical computing one new approach is investigated in this thesis, that is the concept diffractive neural networks. This very new field of research has been accelerated

[1] Quantum computing and optical computing may overlap as quantum computers might use optical waves for entanglement.

by a publication of Lin, Rivenson et al. in 2018 [9]. The corresponding number of publications per year are shown in Figure 1-2.

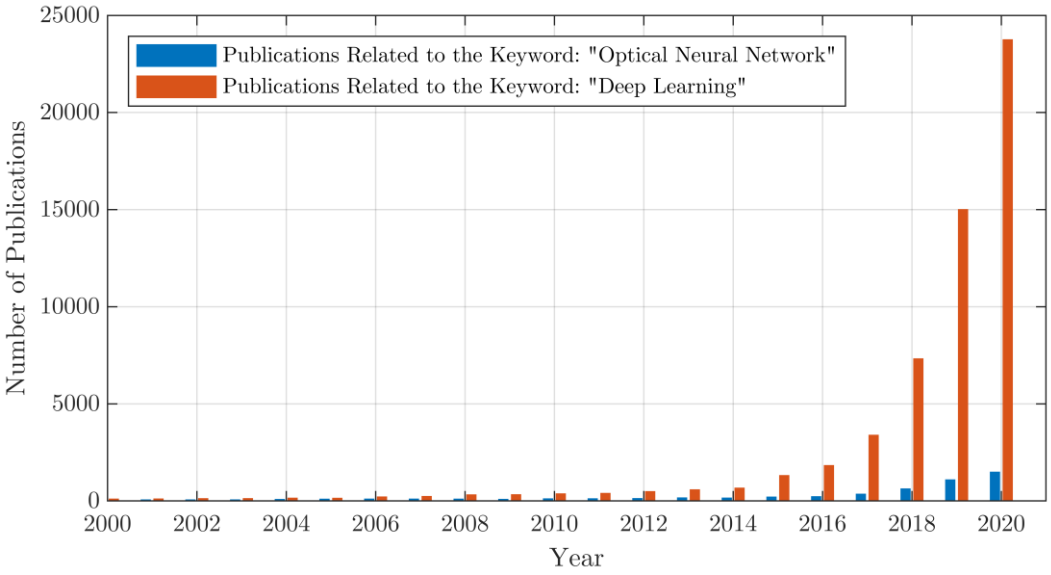


Figure 1-1: Number of publications per year for the search topics of „optical neural network“ in blue and „deep learning“ in orange. The data are taken from web of science [7].

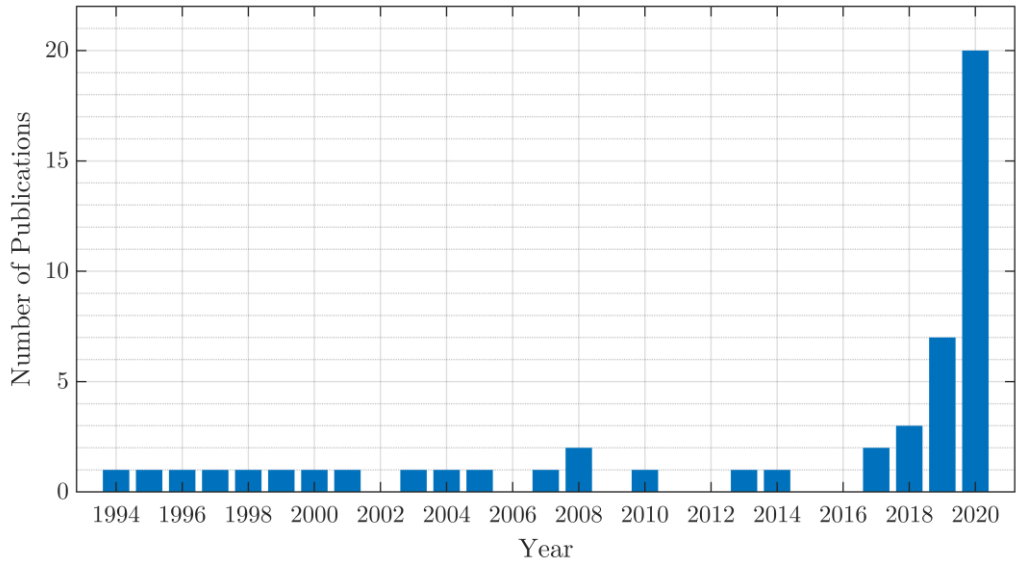


Figure 1-2: Number of publications per year for the search topic of „diffractive neural networks“. The data are taken from web of science [7].

This thesis is structured as follows: Chapter 2 will be an introduction to basic conventional neural networks with emphasis on the methods used in the work of Li, Rivenson et al.. Chapter 3, then reviews current approaches for optical neural computing and transfer methods from conventional neural networks to the optical counter parts. Also, the goals for this work will be derived based on the current state of the art and current challenges as three hypotheses. Chapter 4 is an in-depth derivation of a sophisticated model for calculating the optical field in

diffractive neural networks. The properties of this derived method with regards to sampling is also extensively investigated. Chapter 5 is an experimental section were the methods of chapter 4 are tested with the application to diffractive neural networks in mind and further necessary sampling conditions are made. A summary of this work and an outlook to further research is found in chapter 6 and 7, respectively.

2 Artificial Neural Networks

A main topic of this thesis are optical neural networks. Thus, this chapter is meant to introduce the basics of classical digital or electronic neural networks as a foundation for later discussions. In the first part fundamental definitions of neural networks are given and the forward propagation principle is explained. The second part shows the mathematical derivation of the most common learning algorithm, stochastic gradient descent. Part three then emphasizes on nonlinearity in neural networks and lists common functions to achieve this nonlinearity. The last part gives a quick introduction in a more specialized type of network, that are convolutional neural networks.

2.1 Basic Concept of Artificial Neural Networks

Artificial Neural Networks (ANNs) are based on a network of nodes that process an input set of data points to achieve a specified task. Therefore, each node has parameters associated with it that do a mathematical operation on an input, coming from other nodes. Those nodes are commonly called neurons [10]. ANNs are a subset of the field of machine learning and have been proven to successfully solve tasks in the fields of curve fitting, process control, autonomous driving, pattern and sequence recognition and classification, solving mathematical problems, and many other tasks [11, 12, pp 8, 13–15]. ANNs are inspired by the biological neuron structure in mammal brains, reflecting basic principles. One fundamental principle of ANNs is learning to solve given tasks. Here for, the parameters for each neuron are tweaked iteratively to achieve higher accuracies of the networks' outputs. A trained network may operate independently and apply the learned methods much quicker than other algorithms, due to its smaller holistic complexity[2]. In all further context ANNs will be named NNs, although the "artificial" property is an important distinction, but all references made are to digital, electronic, or optical ANNs.

2.1.1 Neurons in an NN

The most basic unit of a NN is the neuron, named after its biological model. A neuron takes an arbitrary number of input values $x_1 \dots x_j$ and computes an output value y depending on the input. Each input value has a weighting w_j associated with it to mark the relevance of a specific

[2] In a way neural networks can be view as self-organizing algorithms, which uses simpler operations and a straight but unknown path for calculating approximated truths, a.k.a. predictictions.

input to produce a certain output. A neuron has also a bias term b, which is responsible for how easy a neuron is stimulated i.e., apply a threshold for its activation. The activation of a neuron is the output value y. Neuron types are distinguished by their activation function f [6, pp 11–14, 10, 12, pp 11–13]. Formally, a neuron is described by:

$$y = f\left(\sum_j w_j \cdot x_j + b\right) \tag{2-1}$$

A representation of equation (2-1) is shown in Figure 2-1. Where the number of inputs j is three. The input weights are visualized by arrows, each transmitting one input x_j with a "signal strength" or weighting w_j to the neuron itself. The bias term b preloads the neuron to facilitate or impede the neurons activation. The activation function f then relates the sum of all weighted inputs and the bias to the output y.

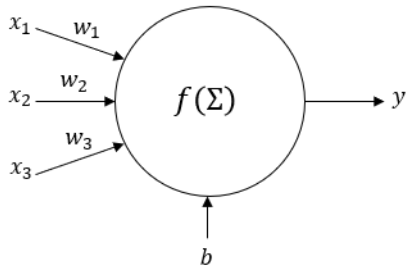


Figure 2-1: Generalized illustration of an artificial neural network neuron with three inputs.

E.g., a binary neuron would be represented by a step activation function:

$$f(z) = \begin{cases} 1, if\ z > 0 \\ 0, if\ z \leq 0 \end{cases} \tag{2-2}$$

With z being the sum of the weighted inputs $z = \sum_j w_j x_j + b$. A neuron with the activation function of equation (2-2) is called a perceptron [16]. A perceptron has a binary output, which means when changing the weighting of the inputs only slightly, the output y might flip from 1 to 0 or vice versa. To achieve a more relatable dependence of the output and weighting factors, a sigmoid activation function might be used.

The sigmoid function is defined as:

$$f(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{2-3}$$

The reason why the sigmoid function is more predictable, lies in the differentiability of the function itself [12, pp 13–18].

By choosing a differentiable function one can predict the change of the neuron output when changing a weight or bias by Δw_j or Δb, respectively with:

$$\Delta y \sim \sum_j \left(\frac{\partial \sigma}{\partial w_j} \Delta w_j + \frac{\partial \sigma}{\partial b} \Delta b \right) \tag{2-4}$$

A more detailed look at activation functions and their properties will be given in section 2.3. How neurons are organized in a basic NN architecture is the topic of the following subsection.

2.1.2 Shallow Artificial Neural Networks

Neurons described in the previous subsection 2.1.1 might solve tasks or computations of more complex nature, for this they have to be organized in a network. The most basic type of network is a feed-forward shallow neural network. Hereby, neurons are ordered in layers. The input layer being the first layer, which receives the input data, one hidden layer and an output layer. The word "shallow" describes the depth of a network i.e., the number of hidden layers in the network. The structure is depicted in Figure 2-2.

The term "feed-forward" defines the direction of information flow through the network[3]. In this case information is transported from the inputs x_j in the left part in Figure 2-2 to the output y in the right part. The input in this case is an array x of four values. The input layer itself does not act on the inputs themselves[4]. The hidden neurons all act according to equation (2-1). Each neuron's output is connected to all neurons of the next layer, indicated by lines. The output layer in the case shown, is a single value $y \in \mathbb{R}$ that might be any real number[5]. By applying a threshold definition to that value by using a binary step activation function in the output layer, the result might be a binary decision output 0 or 1 depending on the input data.

[3] In contrast, there are also recurrent network architectures and architectures with feedback loops which act as a kind of network memory [12, chapter 7].
[4] Although often the same symbol is used, the input layer has no function associated with it, except feeding data into the network.
[5] As shown later, also complex valued networks exist [17].

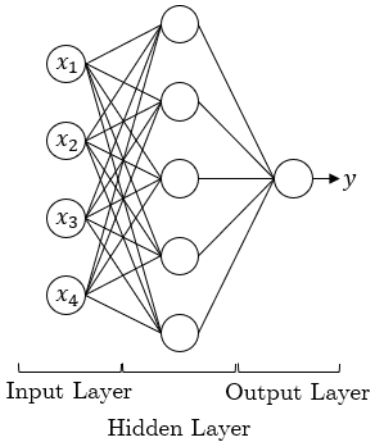


Figure 2-2: Illustration of a shallow neural network with 4 inputs x_j, one hidden layer, and one output layer with one output y.

To describe the inputs and outputs of a neuron inside a networks, equation (2-1) is redefined as follows:

$$a_j^l = f\left(\sum_k w_{jk}^l \cdot a_k^{l-1} + b_j^l\right) \tag{2-5}$$

Hereby a_j^l is the activation, i.e. the output, of the neuron j in the l-th layer of the networks and a_k^{l-1} is then the activation of the neuron k of the layer before. The weight connecting neuron k to neuron j is w_{jk}. The subscripts and superscripts are in respect to the observed neuron. A illustration of the neuron equation (2-5) and the coefficient names are shown in Figure 2-3. The activation of the last network layer is then the network output y. The reverse indexing of the weights may appear confusing, but when using matrix multiplications for calculating z_j^l the indexing ensures that every element of a_k^{l-1} is multiplied with the corresponding element of w_{jk}^l.

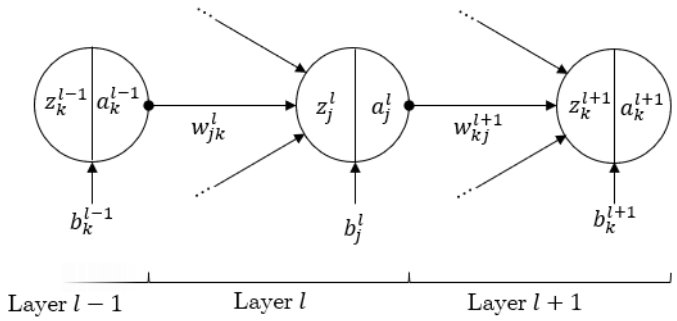


Figure 2-3: Three neurons in a network of layer $l-1$, layer l and the layer $l+1$, with common indexing used for weights, biases, activations.

The specific example of the shallow neural network pictured in Figure 2-2 can be mathematically described now by using the modified equation (2-5) and following the path through the network:

$$
\begin{aligned}
y_j^3 &= f\left(\sum_{k=1}^{5} a_k^2 w_{jk}^3 + b_j^3\right) \\
a_j^2 &= \sigma\left(\sum_{k=1}^{4} a_k^1 w_{jk}^2 + b_j^2\right) \rightarrow a_k^2 \\
x_j &\rightarrow a_k^1
\end{aligned}
\tag{2-6}
$$

The $\rightarrow$ indicates a perspective change to the layer $l+1$ where the respective indices k and j are switched. The input layer in Figure 2-2 has only the identity activation function[6] associated with it and therefore only distributes the input x [6, pp 16–17].

A more complex example would be if one considers a digital image with 28×28 pixels as input, the input vector would be the length of $28^2 = 784$. The given task of the network might be to recognize whether the image shows the number five. The input images might be a set of handwritten numbers from zero to nine. In the given scenario, the architecture of Figure 2-2 would produce a binary output $y \in [0,1]$, assuming the output layer uses the binary step activation function from equation (2-2). Based on this prediction, a decision can be clearly made but if one would redefine the task to decide which number the network perceives, one binary output cannot carry the needed information. The obvious modification that must be made is to increase the number of outputs. Instinctively, one would choose to increase the number of outputs to match the number of objects classes. In the case of ten written digits, ten outputs y_1 to y_{10}.

There are other possible ways to define the output[7] [6, pp 14, 10]. When only increasing the number of outputs, the network might output more than just one positive output or none at all. In both cases, the output cannot be interpreted. Therefore, one other modification must be made. An additional layer is added that uses a softmax function with no weights or biases associated. The softmax layer is defined as[6, pp 14]:

$$
a_j^l = softmax\left(a_k^{l-1}\right) = \frac{e^{a_j^{l-1}}}{\sum_k e^{a_k^{l-1}}}
\tag{2-7}
$$

Where j is the number of the output and k the number of the input neuron. Therefore, the softmax function normalizes the specific output at j with respect to all inputs over k. The

[6] See subsection 2.3 for more details.

[7] E.g. one might encode the class in binary states reducing the number of output neurons to at least three, because: $2^3 = 16 > 10$.

result is that the sum of values over all outputs is one, $\sum y_j = 1$ and the specific output y_j can be regarded as a predicted probability, with values between 0 and 1. Each output will be assigned one specific output class. In case of the set of handwritten digits the input y_1 might be assigned to the number 0, y_2 to 1, and so on.

The last layer, based on the property of predicting the class, is called a classification layer or classifier, as shown in Figure 2-4. In Figure 2-4 the total number of classes is chosen to be three not ten, as in the example of handwritten digits.

With the addition of the classification layer, the network is confined for solving only specific kinds of problems. For explanatory purposes this will do, but keep in mind that changing the activation function of the output layer, not only the output values change but also how they are interpreted [6, pp 11–14].

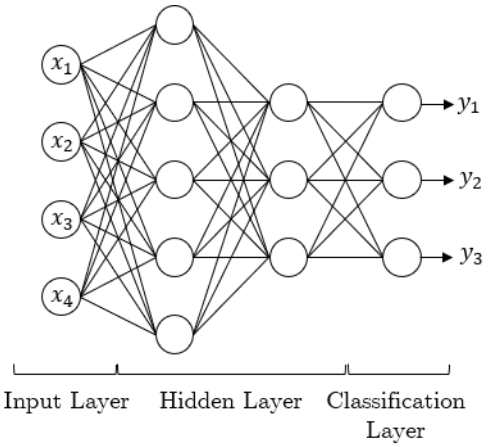


Figure 2-4: Illustration of a shallow neural network with four inputs x_j, two hidden layers, a classification layer and three output classes.

2.1.3 Deep Neural Networks

Often one might stumple over the term „deep neural networks“ (DNN). This term refers to the depth of a network. It can be loosly related to number of hidden layer in a network but ist more an expression for the complextity of a network. For example a NN might have billions of neurons in lots of layers but if their activation functions are linear, the network complexity is actually small, for reasons further explained in section 2.3. DNNs use different types of layers, splitted paths or other methods to solve highly complex tasks. The term „deep“ is therefore only a distinction of task complexity. All basic methods derived in this work can be adopted to DNNs as well. [6, pp 20–21]

2.2 Training of a Basic Neural Network

Now that the basic architecture is established, an introduction into the learning algorithm of NNs will be given in this chapter. Beware, this chapter covers only the basic approach. There are a vast amounts of methods and techniques improving the learning process for different scenarios [6, 18, 19]. Regardless, its irrefutable to get an understanding of the fundamental learning algorithm when working with NNs. This overview contains a quick introduction into two important loss functions. The method for updating the network parameters to achieve an improvement in accuracy is topic of the second subsection and the algorithm for calculating those values is derived in the last part.

2.2.1 The Loss Function

The loss function[8] is a way to evaluate the accuracy of an output y_j of the network. The loss function is the basis of the learning process because it gives a measure of how well the actual output corresponds to a given target, i.e. ground truth [10]. The target t_j is a known correct value of y_j for a known training input. There exist several possible types of loss functions, each with their own application scenarios due to their specific properties and suitable preceding activation functions [6, pp 14–16, 20].

There are basically two types of network targets, regression and classification, which require different loss functions [6, pp 14–16, 10, chapter 3]. If regression values are the targets, the input function produces a continuous output function. An example for regression applications is curve fitting or image denoising. The classification type on the other hand labels input data, like hand-written number recognition. Although the output values might be real, they are predicted possibilities that must be binarized to output a decision. The example given in subsection 2.1.2 in which the recognition of hand-written numbers is the goal, is clearly a classification problem. A typical loss function for classification problems is the cross-entropy loss function [6, pp 14–16]:

$$L_{CE}(n) = -\sum_{j} t_j(n) \cdot \log\left(y_j(n)\right) \tag{2-8}$$

Hereby, y_j is the probability output vector of the network, generated by a softmax layer (see chapter 2.1.2), t_j is the ground truth vector for the respective data input, and j is the number

[8] Sometimes called cost function or objective function.

of the output. Equation (2-8) is the loss function for one input. When training a NN lots of input data are processed to evaluate the networks' performance. To calculate the loss for many input samples, the mean over all data is used:

$$L_{CE} = \frac{1}{M}\sum_{m=1}^{M}\left(-\sum_{j} t_j(m)\cdot\log\left(y_j(m)\right)\right) \tag{2-9}$$

Where M is the number of training samples. A cross-entropy loss is typically combined with a preceding softmax layer due to the probabilistic nature of the cross-entropy function [6, pp 68–69]. A typical classification architecture with three classes is shown in Figure 2-5. The target vector t_j is one-hot encoded, which means that t is zero everywhere except for the entry at j belonging to target output class, which is then one.

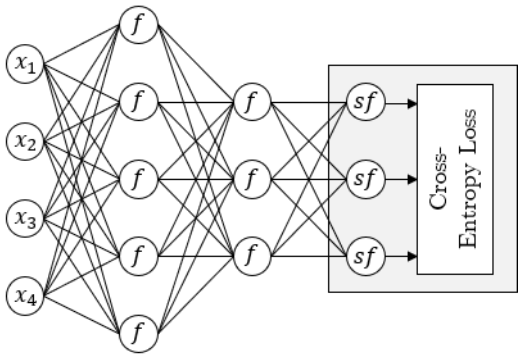


Figure 2-5: Sketch of a typical classification network with three classes. The softmax activation function is abreviated by sf, any other arbitrary activiation function by f.

A typical loss function that can be applied to regression problems is the mean square error (MSE) loss function [6, pp 15, 10]:

$$L_{MSE} = \frac{1}{2M}\sum_{m=1}^{M}\left\|t_j(m) - y_j(m)\right\|^2 \tag{2-10}$$

Hereby, $\|v\|$ is the vector norm of v. In contrast to classification networks, a regression network does not need a specific activation function for the last layer. Often just the identity function is used as an output layer[9]. A typical regression architecture is shown in Figure 2-6.

[9] See chapter 2.3 for more details.

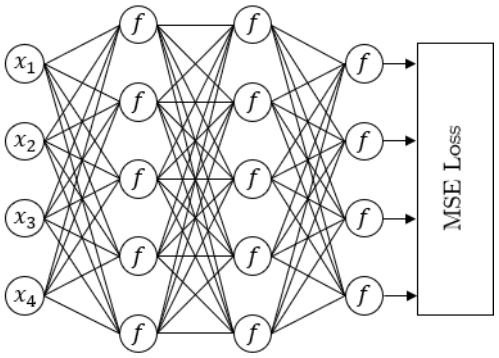


Figure 2-6: Sketch of a typical regression network, where arbitrary activation functions are indicated by f. As loss function the MSE function is used.

Now, that a way for evaluating the network performance is established, the next step is to define a way to change the NN weights and biases to decrease the network loss. A crucial dependence is omitted in equations (2-9) and (2-10), specifically that the loss function depends on the weights, biases and the input itself of the network, although it is implied because the network output y_j clearly depends on the network parameters according to the neurons' description in equation (2-5).

2.2.2 Gradient Descent

Up until this point, it is not clearly established what is meant by training a NN. If we consider an untrained network, all network weights w_{jk}^l and biases b_j^l have no values that particularly produce a specific output y_j with a low loss function. The network must first be trained by data with known outputs. This means, taking a set of labeled data[10], that is exemplary for the intended application and feeding it to the "raw" network. At first, the NN will perform poorly, but when looking at the loss function and how it changes when changing weights and biases one can find a way to tweak the network parameters in the right way so that the network predictions become more accurate. The goal is to minimize the loss function: $\min\left(L\left(w_{jk}^l, b_j^l\right)\right)$. The common algorithm of how the loss function can be minimized is called gradient descent. When speaking of deep neural networks with a vast number of neurons each parameter representing a dimension, the solution space is therefore multi-dimensional as well. For explanatory purposes, the number of dependencies is reduced to one: so that goal is to find the minimum of a one-dimensional function $L(v)$. An example function $L(v)$ is shown in Figure 2-7. Obviously the minimum in the case of one variable might be found by calculus, i.e. using the first and second derivatives of $L(v)$ but with many variables, each representing a dimension,

[10] In case of a classifying network, the ground truth is a data label.

this approach practically will not work [10]. Another approach is to choose a random starting point and change v by a small amount Δv. The value of L will change according to:

$$\Delta L \approx \frac{\partial L}{\partial v} \cdot \Delta v \tag{2-11}$$

Hereby $\partial L/\partial v$ is the partial derivative of L with respect to v i.e., how will the loss function change when v is modified by Δv. As shown in the example in Figure 2-7 if the starting point is v_0 and $v = v_0 + \Delta v$, then ΔL is negative. The value Δv is chosen so that the gradient of the loss function L becomes negative.

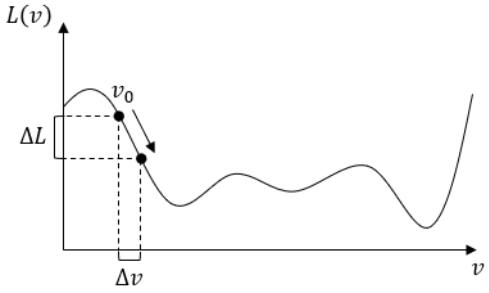


Figure 2-7: An arbitrary function $L(v)$ representing the loss of a network.

Although the example has only one variable[11], equation (2-11) might be redefined using the nabla operator $\nabla \equiv \partial/\partial v$:

$$\Delta L \approx \nabla L \cdot \Delta v \tag{2-12}$$

Here, $\nabla L = \frac{\partial L}{\partial v}$ is the gradient of L. To achieve that ΔL is negative, one can choose Δv to be:

$$\Delta v = -\mu \nabla L \tag{2-13}$$

Where μ is a small factor that is called learning rate. In equation (2-13) it is assured that ΔL is approximatly negative, when μ is small [10 , chapter 1]. The approximation does not holdtrue if μ is large, i.e. larger steps are taken.

The value v can now be uptated repeatedly to find a local minimum of the function $L(v)$:

$$v' = v - \Delta v = v - \mu \nabla L \tag{2-14}$$

This method might be applied to the variables of a NN in the same way. The gradient ∇L becomes $\left(\frac{\partial L}{\partial w_{11}^1} \dots \frac{\partial L}{\partial w_{jk}^l}\right)$ and $\left(\frac{\partial L}{\partial b_1^1} \dots \frac{\partial L}{\partial b_j^l}\right)$.

[11] When L has more than one variable, substituting all partial derivative term by the nabla operator makes cinsideraly more sense than just for one variable.

Further, equation (2-12) becomes $\Delta L \approx \nabla L(\Delta w + \Delta b)$, so that the updateted values can be defined by:

$$w_{jk}^{l}{}' = w_{jk}^{l} - \Delta w_{jk}^{l} = w_{jk}^{l} - \mu \frac{\partial L}{\partial w_{jk}^{l}}$$
$$b_{j}^{l\prime} = b_{j}^{l} - \Delta b_{j}^{l} = b_{j}^{l} - \mu \frac{\partial L}{\partial b_{j}^{l}} \qquad (2\text{-}15)$$

Although it is not defined yet, how the derivative of the loss function L can be calculated with respect to every variable in the network. The gradient descent method gives the means of how to change the values in order to minimize the loss function. But one other more practial problem might be solved right away. That problem is that, for one change in the weights and biases the output of all of the training data has to be calculated. A practial approximation might be made by calculating a small batch of data, then updating the values. Let N be the batch size of the training data set of size M, the gradient vector ∇L might be approximated by:

$$\nabla L = \frac{1}{M}\sum_{m=1}^{M} \nabla L(m) \approx \frac{1}{N}\sum_{n=1}^{N} \nabla L(n) \qquad (2\text{-}16)$$

By taking only a small subset of training data for each updating step, the gradient descent method becomes the stochastical gradient descent method [10 , chapter 1]. This approximation holds true if N is chosen to be sufficiently large, but note that $N \ll M$ should be maintained for compuational efficiency. Figure 2-8 shows how the training data is organized when subdivided into smaller batches. If the training set has M samples and let one batch have N samples then number of batches B becomes:

$$B = \frac{M}{N} \qquad (2\text{-}17)$$

Every time a set batch of size N is processed, all network parameters are updated, and the next subset of all data becomes batch b, and so on. When all samples are calculated, one epoch of training is done. For the next epoch, all data is randomly shuffled, and the steps shown in Figure 2-8 repeated.

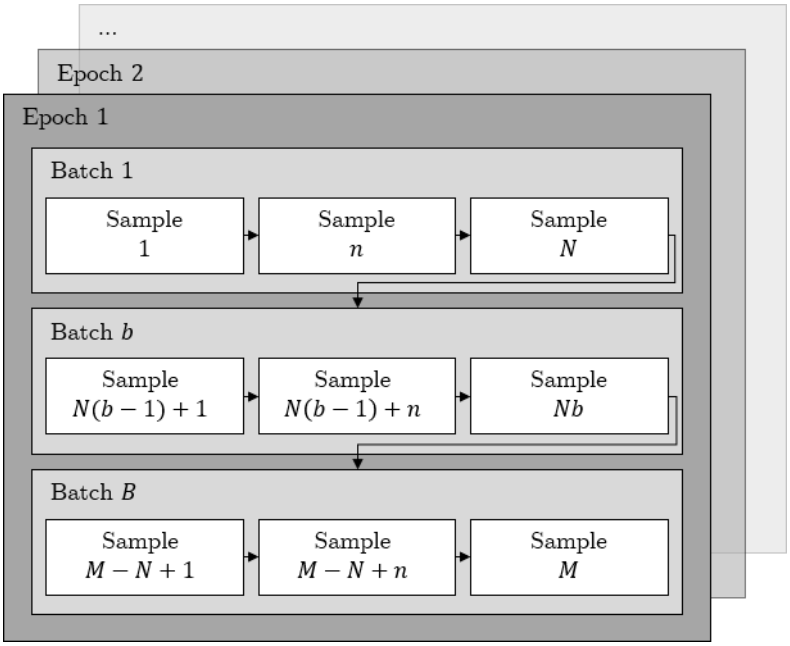


Figure 2-8: Illustration of the first training epoch of M samples organized into smaller batches b, each with N samples.

2.2.3 Backpropagation Model

The backpropagation model was developed independently by at least three persons or groups. In 1981 Werbos [21] wrote the first known modern backpropagation algorithm. In 1982 Rumelhart [22] and Parker [23] also presented the idea of backpropagation. It was not until the, more specific, work of Rumelhart, Hinton and Williams [24] which demonstrated the applications in neural computing, the backpropagation algorithm got attention. What is meant by backpropagation is that by using the chain rule, the error of a network described in the previous chapter 2.1, can be propagated through the network in opposite direction to find the error activation i.e., contribution of each neuron to the overall network error. In this chapter a qualitative explanation of how the backpropagation algorithm works and how it is used to modify the weights and biases in NNs is shown.

The goal is to calculate the partial derivatives $\frac{\partial L}{\partial w_{jk}^l}$ and $\frac{\partial L}{\partial b_j^l}$ from equation (2-15) for each weight and bias. Two intermediate values are defined. The first is based on the equation of a single neurons' activation a_j^l (2-5) and defines the value z_j^l as:

$$z_j^l = w_{jk}^l a_k^{l-1} + b_j^l \rightarrow a_j^l = f\left(z_j^l\right) \tag{2-18}$$

So, z_j^l is just the neurons' activation before applying the neurons' activation function f. The other intermediate value is the error of a neuron δ_j^l [10, chapter 2]. The error of a neuron is a

measure of how much the loss function at the end of the network changes when changing z_j^l. Formally the error is then:

$$\boxed{\delta_j^l = \frac{\partial L}{\partial z_j^l}} \tag{2-19}$$

If $\frac{\partial L}{\partial z_j^l}$ is large, a change Δz_j^l in z_j^l changes the loss function proportionally, but if $\frac{\partial L}{\partial z_j^l}$ is small, changing z_j^l will not change the loss L very much. Therefore, δ_j^l might be seen the error of a neuron, because when $\delta_j^l \approx 0$ the loss function L can not be further minimized. The error δ_j^l can be related to the wanted values $\frac{\partial L}{\partial w_{jk}^l}$ and $\frac{\partial L}{\partial b_j^l}$. This can be achieved using the chain rule. The change in the loss function depending on each weight is then:

$$\frac{\partial L}{\partial w_{jk}^l} = \frac{\partial L}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial w_{jk}^l} = \delta_j^l \cdot a_k^{l-1} \tag{2-20}$$

Where the derivative of equation (2-18) in respect to w_{jk}^l is just the activation of the previous layer a_k^{l-1}. And similar for the bias:

$$\frac{\partial L}{\partial b_j^l} = \frac{\partial L}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial b_j^l} = \delta_j^l \tag{2-21}$$

Where the derivative of equation (2-18) in respect to b_j^l is 1. With this relation established, the errors of each neuron of one layer are calculated beginning at the output layer. Let L be the number of network layers, then the output of the last layer y_j becomes $a_j^{l=L} = a_j^L$. The error of the last layer can be derived by again using the chain rule with equation (2-19) but with respect to z_j^L:

$$\delta_j^L = \frac{\partial L}{\partial z_j^L} = \frac{\partial L}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L} \tag{2-22}$$

Because, by definition from equation (2-18), $a_j^L = f\left(z_j^L\right)$, it follows that $\frac{\partial a_j^L}{\partial z_j^L} = \frac{\partial f\left(z_j^L\right)}{\partial z_j^L} \equiv f'\left(z_j^L\right)$. Equation (2-22) therefore might be written as:

$$\boxed{\delta_j^L = \frac{\partial L}{\partial a_j^L} f'\left(z_j^L\right)} \tag{2-23}$$

Which is the error in the output layer. Next the errors in the networks' hidden layers are derived.

Again, using the chain rule on equation (2-19), but this time for an arbitrary layer and in respect to the next layer $l+1$:

$$\delta_j^l = \frac{\partial L}{\partial z_j^l} = \sum_k \frac{\partial L}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l} = \sum_k \delta_k^{l+1} \frac{\partial z_k^{l+1}}{\partial z_j^l} \tag{2-24}$$

Hereby, $z_k^{l+1}$is shown in Figure 2-9 and defined by:

$$z_k^{l+1} = \sum_j w_{kj}^{l+1} a_j^l + b_k^{l+1} = \sum_j w_{kj}^{l+1} f\left(z_j^l\right) + b_k^{l+1} \tag{2-25}$$

Note that the index k used for indexing in the layers $l-1$ and $l+1$. See Figure 2-9 for indexing relations.

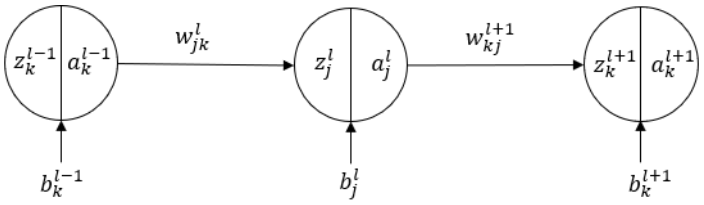


Figure 2-9: Three concatenated neurons with indices. The observation point is neuron j. Only one neuron per layer is shown.

Differentiating z_k^{l+1} in equation (2-25) yields:

$$\frac{\partial z_j^{l+1}}{\partial z_j^l} = w_{kj}^{l+1} f'\left(z_j^l\right) \tag{2-26}$$

Substituting (2-26) into (2-24) results in the error of neuron j in layer l:

$$\boxed{\delta_j^l = \sum_k w_{kj}^{l+1} \delta_k^{l+1} f'\left(z_j^l\right)} \tag{2-27}$$

To find a universal rule for calculation of the error for all layers the equation (2-23) for the last layer and equation (2-27) for hidden layers will be written with slight modifications. According to [10, chapter 2] the equations (2-23) is in vector notation:

$$\delta^L = \nabla_a^L L \circ f'(z^L) \tag{2-28}$$

Where ∇_a^L is the gradient vector $\left(\frac{\partial}{\partial a_1^L} \dots \frac{\partial}{\partial a_j^L}\right)$. And equation (2-27) is then:

$$\delta^l = \left(\left(w^{l+1}\right)^T \delta^{l+1}\right) \circ f'(z^L) \tag{2-29}$$

With (2-28) and (2-29) it can be summarized, that the error might be completely backpropagated by using following rule:

$$\delta^l = \begin{cases} \nabla_a^L L \circ f'(z^l), & if\ l = L \\ \left((w^{l+1})^T \delta^{l+1}\right) \circ f'(z^L), & if\ l \neq L \end{cases} \tag{2-30}$$

To further specify, when using the cross entropy loss function (2-8) its error is according to [25] in the case of multiple classes and a previous softmax layer[12]:

$$\delta_j^L = a_j^L - t_j \tag{2-31}$$

In case of a MSE loss function with a preceding sigmoid activation function layer, the error at the output layer becomes:

$$\delta_j^L = (a_j^L - t_j) f'(z_j^L) = \underbrace{(a_j^L - t_j)}_{\frac{\partial L}{\partial a_j^L}} \cdot \underbrace{\frac{e^{z_j^L}}{\left(e^{z_j^L} + 1\right)^2}}_{\sigma'(z_j^L)} \tag{2-32}$$

Where $\sigma(z_j^L)$ is the sigmoid activation function and $\sigma'(z_j^L)$ its derivative. However which combination of activation functions and loss function is chosen, the error of the last layer is to be calculated first and afterwards each hidden layer. The algorithm for backpropagation can be summarized as follows:

- Calculate the loss of each training sample at the output layer with a loss function L as a measure of performance, e.g. with the cross-entropy loss with equation (2-8) :

  $$L_{CE} = -\sum_j t_j \cdot \log(y_j)$$

  or by using the MSE loss function:

  $$L_{MSE} = \left\| t_j(m) - y_j(m) \right\|^2$$

- Calculate the error of the output with equation (2-28) by:

  $$\delta^L = \nabla_a^L L \circ f'(z^L)$$

  Where the gradient of the loss function in case of the MSE loss function is $\nabla_a^L = a_j^L - t_j$ and f' is the derivation of the last layers' activation. In case of the cross-entropy loss

[12] The derivation of the softmax function is not trivial, but the result is. Therefore, the derivation is not shown here.

function in combination with a softmax layer, the error is directly calculated by equation (2-31): $\delta_{j,CE}^{L} = a_j^L - t_j$.

- Calculate the error of each layer $L-1$, $L-2$, ... with equation (2-29) by:

$$\delta^l = \left(\left(w^{l+1}\right)^T \delta^{l+1}\right) \circ f'(z^L)$$

Where δ^{l+1} is the error of the previous layer.

- Calculate the gradients for the stochastic gradient method from equation (2-20) and (2-21) for the weights and biases respectively by:

$$\frac{\partial L}{\partial w_{jk}^l} = \delta_j^l \cdot a_k^{l-1}$$
$$\frac{\partial L}{\partial b_j^l} = \delta_j^l$$

- Calculate the mean update values $\overline{\Delta w_{jk}^l}$ and $\overline{\Delta b_j^l}$ with equation (2-15) over all batch data samples by:

$$\overline{\Delta w_{jk}^l} = \frac{\mu}{N} \sum_{n=1}^{N} \frac{\partial L}{\partial w_{jk}^l}$$
$$\overline{\Delta b_j^l} = \frac{\mu}{N} \sum_{n=1}^{N} \frac{\partial L}{\partial b_j^l} \qquad (2\text{-}33)$$

- Update the network parameters by:

$$w_{jk}^{l\,\prime} = w_{jk}^l - \Delta w_{jk}^l$$
$$b_j^{l\prime} = b_j^l - \Delta b_j^l$$

- Repeat until satisfactory.

2.3 Activation Functions and Nonlinearity

In this chapter, some common activation functions are listed, explained, and their derivatives are shown. They are divided by their linearity property. A qualitative explanation is given why neural networks need nonlinear components. First, two linear activation functions are presented in the following:

- Binary step activation function:

  The step activation function is shown in Figure 2-10 and defined by (from equation (2-2)):

$$f(z) = \begin{cases} 1, if\ z > 0 \\ 0, if\ z \leq 0 \end{cases}$$

  Its derivative $f'(z)$ is everywhere zero except at $z = 0$ where it is infinite. The binary step function applies a threshold to the input x and outputs a decision, either 0 or 1. It might be used as an output of a binary classifier instead of the softmax function. The gradient $f'(x)$ of the binary step function has no real valued relation to the input value x, which is a problem when calculating the error of a neuron δ_j^l, that depends on $f'(x)$ and therefore is practically always zero.

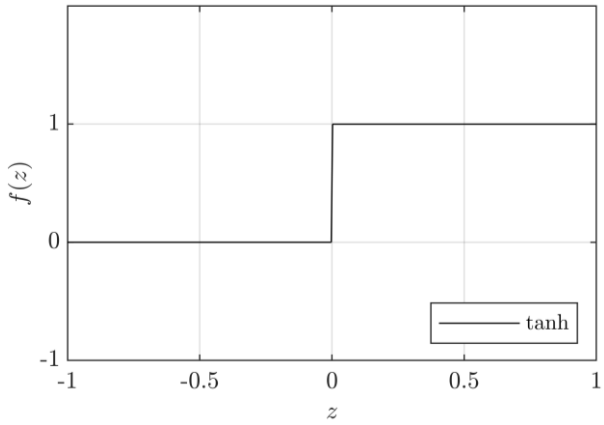


Figure 2-10: Binary step activation function.

- Identity or linear activation function:

  The identity activation function, a.k.a. linear activation function shown in Figure 2-11, relates the input x to the output by:

$$f(z) = z \tag{2-34}$$

  The identity function translates the input to the output with no modifications. The derivative of the identity function $f'(x)$ is everywhere 1. The identity function is by definition a linear function, when one chooses two adjacent layers with both layers only having linear activations, the mathematical operation of both might be also expressed in one single operation. Multiple layers using identity activation functions collapse into one single layer, proven in [6, pp 30–32]. Using only identity activation functions makes not much sense. The identity function can be applied as an output layer function in the case of binary classification problems or in case of regression tasks. In the first case, the

solution space is transformed by any nonlinear activation (shown below) to be linearly separable. This solution space can be properly divided by a linear function [6, pp 32–34]. In the second case, the identity function is used for a linear summation of non-linear transformed inputs. So, for curve fitting a linear output layer is absolutely viable when a non-linear layer precedes [12, pp 86–90].

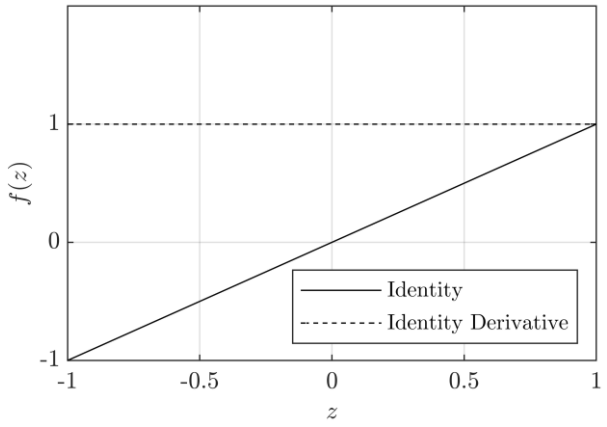


Figure 2-11: A linear activation function $f(z) = z$ and ist derivative $f'(z) = 1$.

Now, in case of nonlinear activation functions three types are shown in the following list. Nonlinear activation functions solve the problem of the two activation functions above. That are differentiateability in case of the binary step function and the non-existant relation of the derivativs on the input itself. The nonlinear activation allows deep learning because multiple layers of those functions can transform the input space. The transformations become more powerful when more layer and neurons are used [6, pp 30–32, 19, pp 120–125].

- Rectifing Linear Unit (ReLU) ativation function:

The ReLU activation function and its derivative are shown in Figure 2-12. The ReLU function is defined as:

$$f(z) = \begin{cases} 0\,, if\ z \leq 0 \\ z\,, if\ z > 0 \end{cases} \equiv \max(0, z) \tag{2-35}$$

And its derivative is defined as:

$$f'(z) = \begin{cases} 0\,, if\ z \leq 0 \\ 1\,, if\ z > 0 \end{cases} \tag{2-36}$$

The ReLU activation limits the possible value of $f(x)$ to positive values, above zero its characteristic is linear. When training a network, the fact that the derivation of the ReLU function is zero below values of zero, can cause a problem called dying ReLU [6, pp 133–134], in which weights do not get updated anymore because the neurons‘ value

is far below zero. This problem might be dealt with a simple modification in which the function becomes a leaky ReLU [6, pp 133–134], but this will not be a topic here.

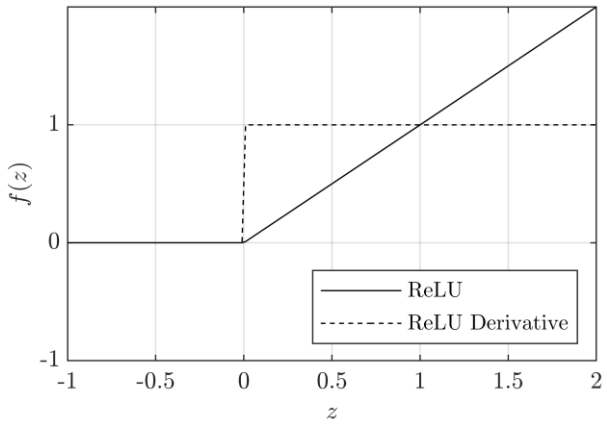


Figure 2-12: Rectifing Linear Unit (ReLU) activiation function and its derivative

- Sigmoid activation function:

  The sigmoid function was first introduced in chapter 2.1.1 in equation (2-3) as an example activation function [6, pp 16–17] is:

$$\sigma(z) \equiv f(z) = \frac{1}{1 + e^{-z}}$$

  And its derivative is defined as:

$$\sigma'(z) \equiv f'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = \sigma(z)\big(1 - \sigma(z)\big) \tag{2-37}$$

  Both, the function and ist derivative are shown in Figure 2-13. The sigmoid function is a normalizing function, which menas that all vaues of $\sigma(z)$ are in the interval $[0,1]$. Also, the function is not-zero centered, which means negative values will be transformed into positive values.

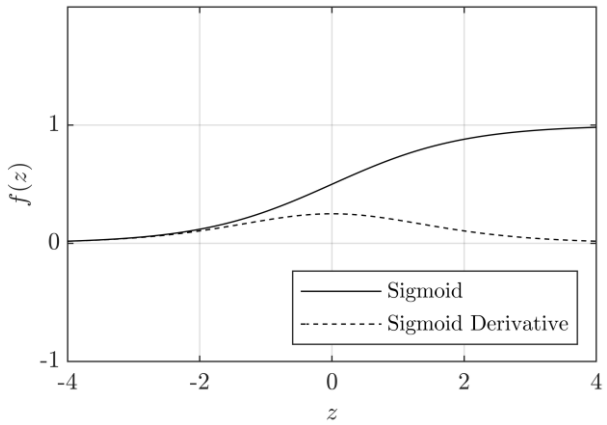


Figure 2-13: Sigmoid activation function $\sigma(z)$ and its derivative.

- Hyperbolic tangent (tanh) activation function:

  The tanh activation function is defined by:

$$tanh(x) \equiv f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{2-38}$$

  And its derivative is:

$$tanh'(x) \equiv f'^{(x)} = \frac{4e^{2x}}{(e^{2x} + 1)^2} = 1 - \tanh^2(x) \tag{2-39}$$

  The tanh activation function is quite similiar to the sigmoid activation function with the difference of being zero-centerd and the derivative being more steep than the sigmoids‘.

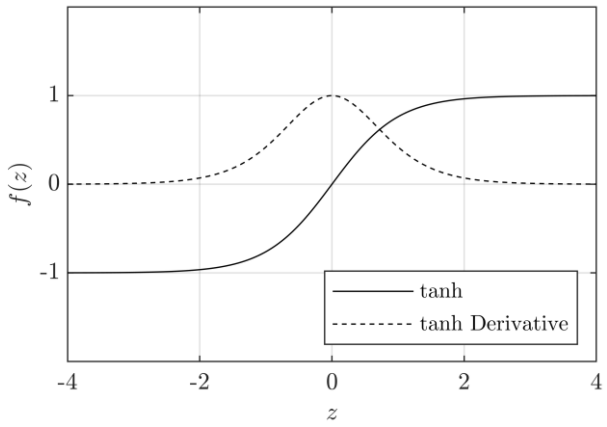


Figure 2-14: The hyperbolic tangent (tanh) activation function and its derivative

The choosing of an activation function lies in the nature of the problem at hand. Linear activation functions might only be applied to linear problems or linearized problems. Most tasks do not have linearly seperable solutions, therefore nonlinear functions are necessary to do classicfication or nonlinear regression. In case of regression tasks, the network can only apply linear regression to a function.

A simple classification example from [6] is shown in Figure 2-15 a) to c). The input space x_1, x_2 is shown in Figure 2-15 a), where three inputs ($(-1,1)$, $(0,1)$ and $(1,1)$) of two classes (A, B) are not linearly seperable. That means that no line can be drawn to divide the two classes correctly. A network consisting of only linear activations can not correctly predict each class[13]. A ReLU layer transforms the solution space of the intermediate values to a graph as shown in

[13] With an increasing number of neurons, eventually the network might work as intended using only linear functions, but with increasing dimensions in the solution space and task complexity the linear neurons needed is vast.

Figure 2-15 b). A network suitable for the transformation task can be build by using one ReLU layer consisting of two neurons and a linear output neuron, shown in Figure 2-15 c). When one of three inputs is fed into the network, the weights indicated as numbers on the lines connecting the input neurons to the hidden neurons, are multiplied with the input and all inputs to a neuron are summed up according to equation (2-5).

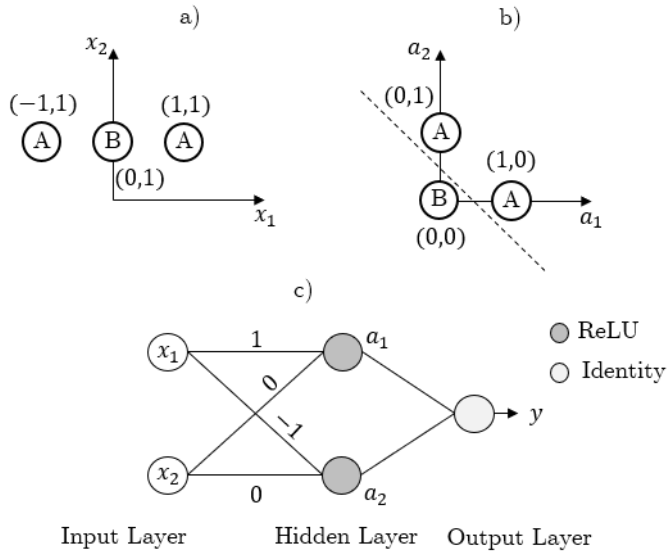


Figure 2-15: Example of the solution space transformation by a layer of ReLU activation function. In a) the three inputs can not be divided by linear regression. In b) the transformed solution space at $a_{1,2}$ is shown. Now, a line can be drawn to seperate the classes A and B. In c) the network performing this transormation is shown.

The weight matrix of Figure 2-15 c) can be written as:

$$w_{jk} = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}$$

And the three inputs as:

$$x_j(1) = \underbrace{\begin{pmatrix} -1 \\ 1 \end{pmatrix}}_{A}, x_j(2) = \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{B}, \underbrace{x_j(3) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}}_{A}$$

The hidden layer is calculated then for the first input as follows:

$$a_k(1) = w_{jk}x_j = ReLU\left(\begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} -1 \\ 1 \end{pmatrix}\right) = ReLU\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

For the second input it is:

$$a_k(2) = ReLU\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

And finally for the third input the hidden layer is:

$$a_k(3) = ReLU\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Now, the classes may be separated by a linear regression neuron, indicated as a dashed line. This linear neuron would be the single output neuron. So, nonlinear activation functions transform the solution space to be linearly seperable and this is what gives NNs their computational power [6, pp 32–34, 12, pp 32–33, 26]. Vaster and bigger networks can perfom much more complex transformations and solve much more complex tasks.

2.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) are often applied to data in the form of images. This image data is usually arranged in a two-dimensional grid. The convolution is by its fundamental principle well suited for correlating spatial dependencies. The motivation for CNNs mostly comes from image classification tasks, like the handwritten number example from chapter 2.1. Spatially encoded features of an image may be processed by a CNN without the necessity of being spatially shift-invariant, that means for example the number nine might be recognized in an image regardless of its position in the image. To achieve this task the convolution is utilized. Let u and h be two two-dimensional vectors of arbitrary size, then the discrete convolution[14] of both matrices g is:

$$g(n,m) = \sum_{l=1}^{L} \sum_{k=1}^{K} h(l,k) \cdot u(n-l, m-k) \Leftrightarrow g = u * h \tag{2-40}$$

Hereby h is called a filter or kernel of size $L \times K$ and is usually smaller than the data matrix u it is operating on. The convolution is abbreviated by the $*$-operator in all further mentions. The convolution can be visualized by the kernel h moving over u, and at every point n,m the coefficients in the kernel are elementwise multiplied with the values in u at that position of h and then summed up to one value. The convolution is also shown in Figure 2-16, where the smaller kernel h moves over the data u.

As it becomes clear from Figure 2-16, the edges of g cannot be reached by the kernel without leaving the data field u. The convolution result g would be smaller than u. To avoid this problem the matrix u can be padded with zeros at the edges[15] and the resulting matrix u cropped by exactly this padding size.

[14] See subsection 4.1.5 and 4.2.1 for more details on discrete convolutions.

[15] The padding size has to be half the kernel size when the size is even, when it is odd the padding size is half minus one half.

Common convolution algorithms usually do the padding and give options for only returning the smaller valid matrix size [27]. However, important is that a specific kernel might be chosen to do specific calculations on the image, like edge filter, blurring, sharpening, and others [28]. These kernels are relatively small but extract specific spatial dependent features. An example is shown in Figure 2-17 where two sobel filters [29] are used to extract edges from an image.

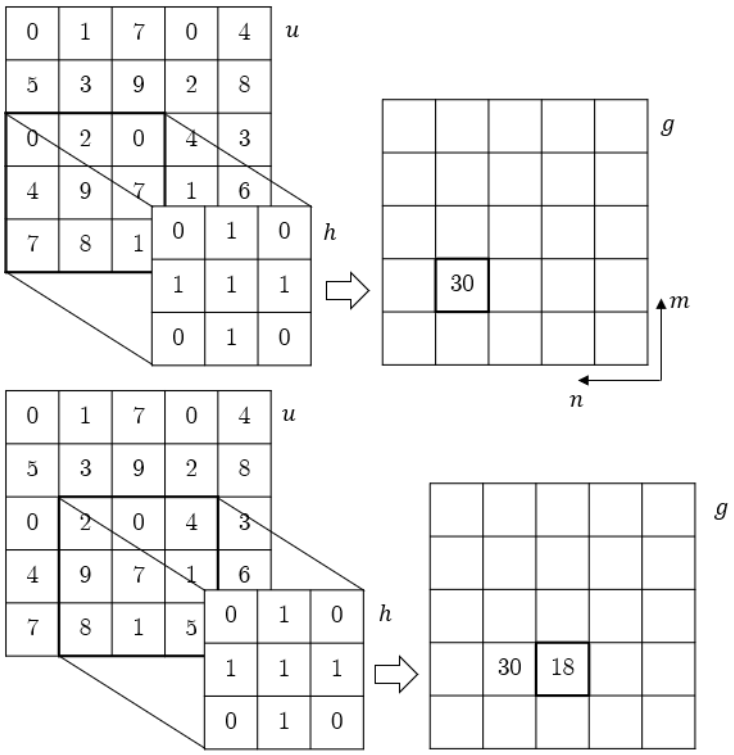


Figure 2-16: Representation of a convolution of the kernel h with a matrix u. The result is the matrix g. Here shown are only two calculation steps for the valid region.

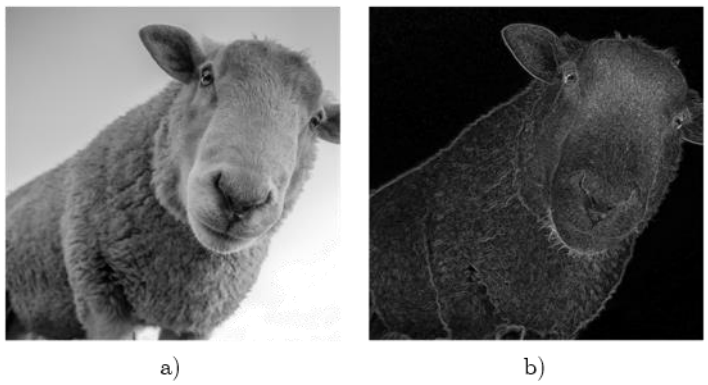


Figure 2-17: Image of a sheep a), applied sobel filter in x- and y-direction to visualize the gradient of the grayscale values in vertical direction b).

In CNNs trainable kernels learn to find features of objects in an image, which are not restricted to only detect edges but may isolate specific geometric properties of an object independent of the location in the image. Typically, several kernels are used in one convolution layer. Each kernel trying to isolate a specific feature of the same image. When applying a stacked kernel

to a two-dimensional image, equation (2-40) must be modified just by an additional stacking dimension d, like:

$$g(n,m,d) = \sum_{l=1}^{L}\sum_{k=1}^{K} h(l,k)\cdot u(n-l,m-k,d) \tag{2-41}$$

The resulting matrix has the size $N \times M \times d$.

Usually a convolution layer is accompanied by a ReLU layer [6, pp 325] and a pooling layer [6, pp 326–327]. The pooling layer, often also called subsampling layer, refers to an operation acting on the convolved layer. In a rectangular subsection the maximum or mean value of that section is used to reduce the size of the layers in n- and m-dimension. Typically, the input image is convolved and subsampled until one activation value per detected feature is left.

The number of features for specific tasks is established when building the network and is chosen by the task at hand. The list of feature activations is then fed into a feed-forward network described in chapter 2.1. The feed-forward network may use linear activations to transmit the weighted feature to a softmax classification output layer. One of the first convolutional neural networks is called LeNet-5 [30],which was trained to recognize hand-written numbers. A sketch of the architecture from Lecun et al. used for LeNet-5 is shown in Figure 2-18. The input is condensed by five convolution layers from a 32×32 pixel image to 120 features that are evaluated by the feed-forward network, called full connection (FC) layer in Figure 2-18.

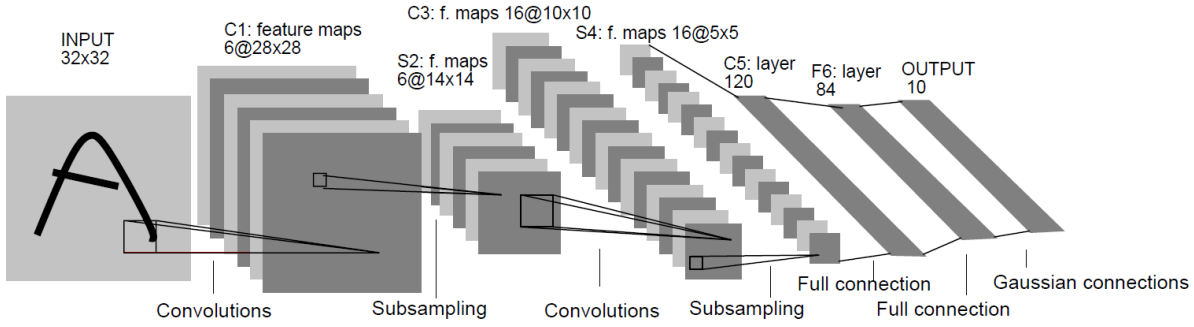


Figure 2-18: Architecture of LeNet-5 a convolutional neural network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.[30]

Convolutional networks have tremendous advantages over classical networks shown in section 2.1, because of their ability to generalize feature locations inside an image. Furthermore, the convolution is a computational efficient when using a discrete type of Fourier transformation[16].

[16] see chapter 4.2.1 for more details

Formally, the forward pass through a convolutional layer may be described by modifying equation (2-5) to:

$$z_j^l = f\left(\sum_k a_k^{l-1} * k_{kj}^l + b_j^l\right) \tag{2-42}$$

Where k is the number of the input layer[17], and j the number of the output map. These indices are used in another way than in the definitions for feed-forward networks from chapter 2.1, because it is assumed that the relative location of each image point and its transformations is preserved in the two-dimensional image grid. Regardless, the kernel k_{kj}^l maps the input a_k^{l-1} to the output map z_j^l. To each output map a bias b_j^l is added. Note that the convolution operation used here uses only the valid region of the output map z_j^l. That means no padding is used and z_j^l will be smaller by the size of the kernel, if the kernel size is even, if it is odd than z_j^l is smaller by the kernel size minus one. The pooling layer is defined by:

$$a_j^l = down(x_j^l) \tag{2-43}$$

Where $down()$ is a down sampling operation, that takes the mean value over a quadratic block with the size $n \times n$ in the layer x_j^l or in case of a max pooling layer the maximum value over the block. The map x_j^l is calculated by applying an activation function to z_j^l, so that $x_j^l = f(z_j^l)$.

It is not directly obvious how one may apply the backpropagation algorithm to a CNN. Here it is assumed that a convolutional layer is followed by a pooling layer, as shown in Figure 2-18. According to [31] the error δ_j^l of a convolution layer is, when the next layer $l+1$ is also a convolution layer:

$$\delta_j^l = up(\delta_j^{l+1}) \circ f'(z_j^l) \tag{2-44}$$

Where $up()$ is a upsampling operation. When upsampling one must pay attention to which pooling method was applied. In case of mean pooling the error values δ_j^{l+1} are uniformly distributed to fill the upsampling blocks of $n \times n$ points and are divided by $\frac{1}{n^2}$. In this case the sampling factor would be n. If the max pooling method is used, then the error is zero for all values inside the upsampling block except for the point which had the maximum value in the

[17] For a RGB image k would run from 1 to 3.

forward pass. At this point the error is δ_j^{l+1}. If the next layer is a fully connected feed-forward layer, then the error δ_j^l of the convolution layer at l is:

$$\delta_j^l = up\left(\left(w_{kj}^l\right)^T \delta_j^{l+1}\right) \circ f'\left(z_j^l\right) \tag{2-45}$$

Where w_{kj}^l are the weights of the feed-forward layer and f' is the derivative of the ReLU layer defined in section 2.3.

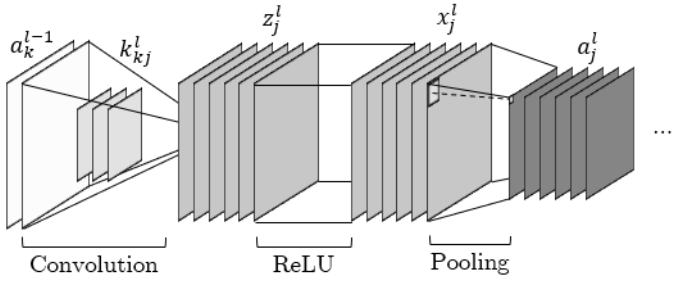


Figure 2-19: A convolution layer and a pooling layer in a CNN. The input a_k^{l-1} with two channels k is convolved with the three kernels k_{kj}^l to produce the six maps z_j^l. Not shown here is the addition of the bias b_j^l to each channel in z_j^l. A ReLU layer is applied tor produce x_j^l with the same channel count, which is then pooled to produce the output map a_j^l, also with the same channel count.

To calculate the gradient vector of the kernel weights $\nabla k_{kj}^l = \partial L / \partial k_{kj}^l$ the following rule is applied:

$$\frac{\partial L}{\partial k_{kj}^l} = \sum_u \sum_v \delta_j^l(u,v) \cdot \left(p_k^{l-1}\right)_{u,v} \tag{2-46}$$

Where $\left(p_k^{l-1}\right)_{u,v}$ is the patch in a_k^{l-1} that was multiplied by the kernel to calculate the value at u and v in the map z_j^l. Because every value in the kernel has an influence on every pixel u and v in δ_j^l, the sum over u and v is taken.

When using matrix multiplications the equation (2-46) can be also be calculated by using following relation [31]:

$$\frac{\partial L}{\partial k_{kj}^l} = rot_{180°}\left(a_j^{l-1} * rot_{180°}\left(\delta_j^l\right)\right) \tag{2-47}$$

Where $rot_{180°}()$ is a rotation of 180° of each two-dimensional map. For a qualitive explanation why the backpropagation uses a rotated filter kernel, it is referred to [6, pp 334–337]. The gradient vector of the bias $\nabla b_j^l = \frac{\partial L}{\partial b_j^l}$ is calculated according to [31] by:

$$\frac{\partial L}{\partial b_j^l} = \sum_u \sum_v \delta_j^l(u,v) \tag{2-48}$$

The gradients ∇k_{kj}^{l} and ∇b_{j}^{l} are used equivalent to equation (2-33) to update the kernel coefficients and biases of the respective layer.

To summarize this chapter: Convolutional neural networks are especially useful for image classification and pattern recognition in images, as they consider the spatial proximity of data by using convolutions. CNNs also can classify shift variant patterns by using pooling. The "vanilla" backpropagation algorithm of subsection 2.2.3 can be applied with the modifications above. A CNN is commonly combined with (linear) feed-forward network that realizes the classification part of the features extracted by the CNN.

3 Optical Neural Networks

3.1 State of the Art

The idea of optical computing has been around for over half a century now. Goodman proposed a theoretical model of an optical vector-matrix multiplier in 1978 [32, pp 285–290]. Since then, a lot of progress has been made in computing, NNs and optical computing. Following the proposal of Goodman, Farhat et al. reported in 1984 [33] an optical implementation of a vector-matrix multiplier and neural network based on the Hopfield model [34], which is network with feedback loops. Using early integrated photonic circuits, Ohta et al. presented an optical synaptic chip with 32 neurons in 1989 [35]. 1993 Kuratomi et al. reported an optical neural network with two hidden layers using micro lens arrays and recognizing hand-written digits [36] with a prediction accuracy of around 80 %. The first optical neural network utilizing holography was reported by Psaltis et al. in 1990 [37]. This network was capable to learn using exposure of photorefractive crystals used in classical holography and imprinting the networks' coefficients into the material. Although, NNs as optical implementations are almost as old as the idea of digital NNs themselves, one rarely encounters a physical device capable of computing using optical processor units, even in scientific applications. The advantages using photons instead of electrons lie for one in massive connection parallelism. The photon velocity is literally light speed, which then also is the maximum speed of which information is transported, that can be realized with current technology[18]. Furthermore, optical systems are almost immune to electro-magnetic interference and neural connections do not interfere with each other, as in high density processors in which crosstalk is a major problem. An additional promise of optical computing is higher energy efficiency in comparison to classical computing [39]. What keeps optical neural networks from revolutionizing the field of digital computing is for one the still relatively large size, with relatively low neuron density, and the realization of nonlinearity in network nodes. These drawbacks might be solved by solutions using integrated photonic circuits, but those systems are complex and expensive to build. Regardless, of which system is implemented, an electronical backend is always necessary by todays technological possibilities. Up until now several approaches of optical neural networks have become promising technologies for realizing practical devices.

[18]Although experimental setups for information transport using quatum teleportation exist [38], this technology is not considered here.

A complete overview covering current advances in optical neural networks dating to 2019 is done by de Marinis, Castoldi and Andriolli [8]. Figure 3-1 is taken from their work and shows their categorization of optical neural network, or as they call it: photonic neural networks (PNNs). De Marinis et al. categorize optical neural networks by their inherent capability of physical memory, which they denote as stateless or stateful PNNs. Covering all types of optical neural networks would extend beyond the scope of this work, but a few major technologies reported in recent years will be shortly mentioned in the remains of this chapter. Additionally, the work will focus on the stateless type of optical neural networks. All publications belonging to each technology can be taken from the original publication of de Marinis et al. [8].

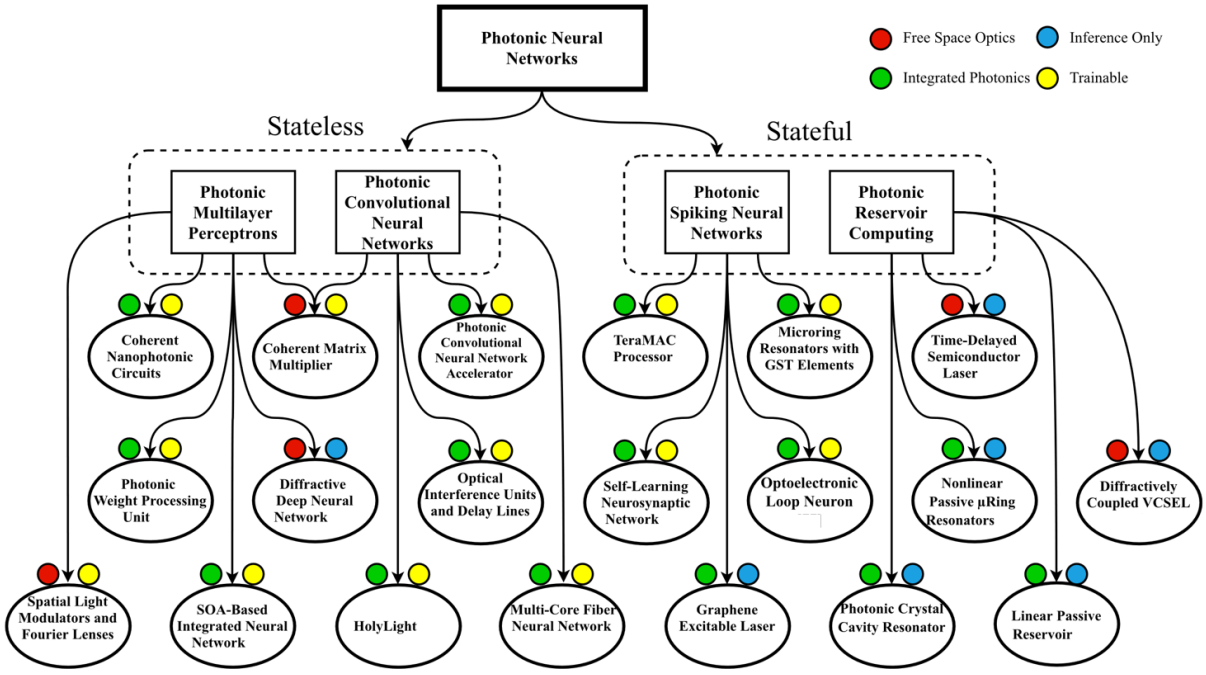


Figure 3-1: Taxonomy of PNN approaches and associated proofs of concept, indicating the hardware implementation (free space optics or integrated photonics) and the operation mode (inference only or trainable). Only the types of neural networks for which a photonic version has been demonstrated in the literature are reported. [8]

One approach getting attention in recent years due to major advances in the field are integrated photonic neural networks. Integrated photonics use optical waveguides to "rebuild" classical computing by reinventing components using only optical signal processing. In 2017 Tait et al. from Princeton University reported a photonic neural network on silicon and demonstrated its computational power by solving differential equations [40]. An illustration of their experimental setup is shown in Figure 3-2. According to [8] this setup might be categorized as a photonic convolutional network accelerator. It relies on dense wavelength division multiplexing (DWDM), so it realizes parallelism through wavelength channels.

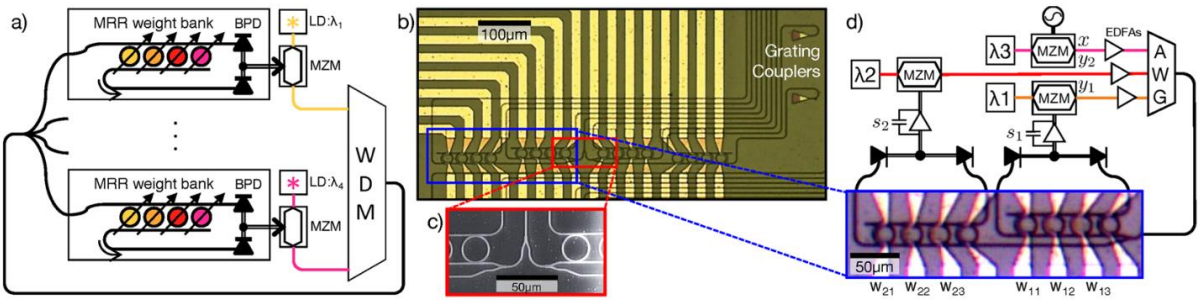


Figure 3-2: Broadcast-and-weight protocol and experiment of Tait et al.. (a) Concept of a broadcast-and-weight network with modulators used as neurons. MRR: microring resonator, BPD: balanced photodiode, LD: laser diode, MZM: Mach-Zehnder modulator, WDM: wavelength-division multiplexer. (b) Micrograph of 4-node recurrent broadcast-and-weight network with 16 tunable microring (MRR) weights and fiber-to-chip grating couplers. (c) Scanning electron micrograph of 1:4 splitter. (d) Experimental setup with two off-chip MZM neurons and one external input. Signals are wavelength-multiplexed in an arrayed waveguide grating (AWG) and coupled into a 2×3 subnetwork with MRR weights, w_{11}, w_{12}, etc. Neuron state is represented by voltages s_2 and s_2 across low-pass filtered transimpedance amplifiers, which receive inputs from the balanced photodetectors of each MRR weight bank. [40]

Simultaneosly, researchers at MIT developed an optical neural network based on Mach-Zehnder interferometers (MZIs) on silicon [41]. In contrast to the solution by Tait et al. does the group around Shen and Harris used a monochromatic approach. They experimentally demonstrated their device by training the network to recognize vowel sounds in speech recordings with a prediction accuracy of 76.6 % and later on they achieved an accuracy of 95 % on the MNIST hand-written number data set [42]. An illustration of their neural network and one optical interference unit (OIU) is shown in Figure 3-3 a) and b) respectively. Noteworthy is that in [41] the nonlinear part was only implemented in sumulations but not in the actual experiments. The implementation of nonlinear components is already confirmed by using saturable absorber materials, like graphene, [43] or bistable optical switches by using photonic crystals [44]. The weights are applied by the tunable MZIs. This type of network would be categorized as a coherent nanophotonic circuit, according to [8].

Although, integrated photonic networks are one of the more promising solutions to realize complex optical networks, they require highly sophisticated manufacturing equipment and technological advances have only been made in recent years, so the advantages of this technology have not been explored to a major extend. Many publications still simulate nonlinearities in software, due to the cumbersome implementation or implement critical parts as electronic components [8].

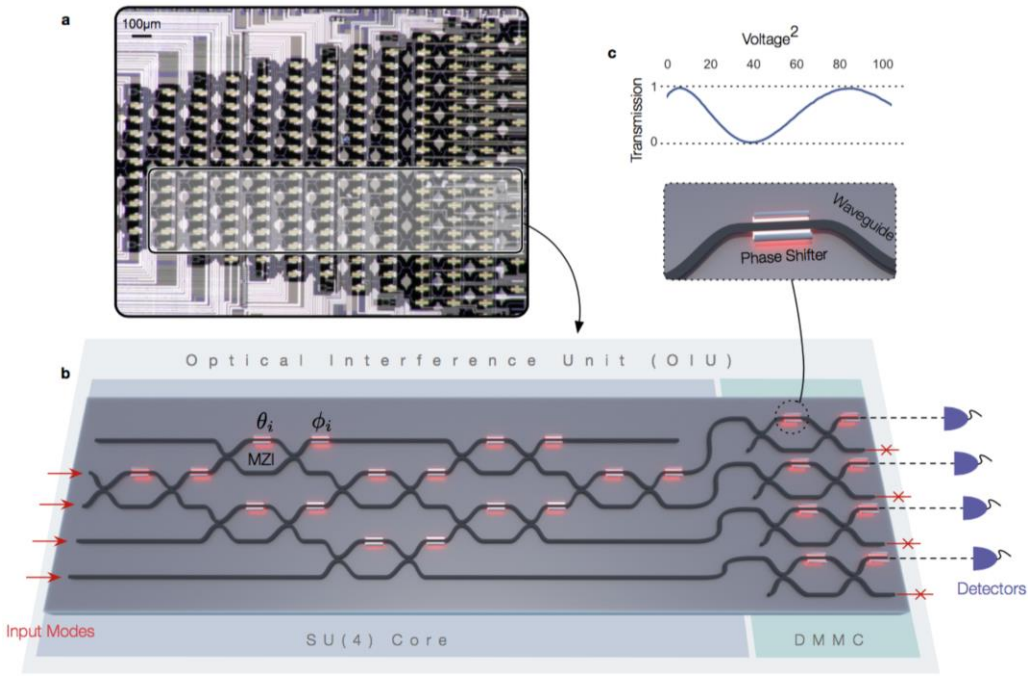


Figure 3-3: Illustration of Optical Interference Unit a. Optical micrograph of an experimentally fabricated 22-mode on-chip optical interference unit; the physical region where the optical neural network program exists is highlighted in grey. The system acts as an optical field-programmable gate array–a test bed for optical experiments. b. Schematic illustration of the optical neural network program demonstrated here which realizes both matrix multiplication and amplification fully optically. c. Schematic illustration of a single phase shifter in the Mach-Zehnder Interferometer (MZI) and the transmission curve for tuning the internalphase shifter of the MZI. [41]

Another approach of realizing optical neural networks is by using free-space propagation and diffraction of optical modes. A proposal made by Zuo et al. in 2019 realized a two layer optical neural network with nonlinear activation [45]. The linear operation is implemented using spatial light modulators (SLMs) and Fourier lenses. The nonlinear operation is realized by using laser-cooled atoms with electromagnetically induced transparency. With an experimental setup, magnitudes larger than the integrated photonic counterpart and no real scalability. This approach is no real alternative, but the usage of a new type of nonlinearity shows potential. The experimental setup is shown in Figure 3-4 a). The data fed to the network is encoded onto a SLM, which therefore acts as the electronic-optic interface. This type of coupling usually used in experimental setup utilizing free space propagation. This approach can be affiliated to the category spatial light modulators and Fourier lenses in the overview of [8].

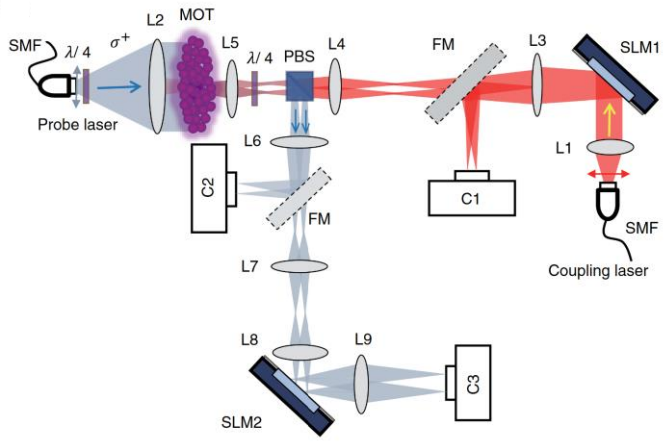


Figure 3-4: Fully functioning two-layer AONN experimental setup of Zou et al. with a nonlinear element implemented as using laser-cooled atoms with electromagnetically induced transparency indicated as MOT[45].

The last approach presented here is also based on free space propagation but uses only passive optical components. Presenting it as a diffractive deep neural network (D^2NN), Lin et al. published 2018 their work in nature [9]. The approach hereby is to use diffractive layers which have complex modulation information imprinted on them. Each layer acts as a linear matrix multiplier. The interconnection of neurons is realized through the coupling of each neurons' wavelet in free space between adjacent diffractive layers. The multiple diffracted field interferes behind the diffraction layers on a detector. The interference pattern is the superposition of all wavelets and is measured as an intensity. The design was demonstrated as an optical neural network classifier and an imager applying regression for image correction. In the case of the classification application specific regions in the detector plane are assigned to one specific output class. So, the relative intensity of one area in respect to all others gives a relative prediction of the input image. In case of the regression task, the intensity image is regarded as a real two-dimensional function and output target is an aberration or disturbance-free image. In later work [46] they achieved around 97 %[19] accuracy in the MNIST hand written digit recognition task. In contrast to the solutions above this approach uses a fixed network design. The network training is done in a traditional computer and then transferred into a network layer design which then was 3D-printed. By using the static approach with only passive components new areas of applications became possible. Lin et al. proposed the application as an integrated component for intelligent sensors. Therefore, they also proposed opto-electrical hybrid networks with the capability of learning to a certain extend [46]. The drawbacks of this technology are for one, it has only been experimentally tested using terahertz wavelength,

[19] By using an opto-electrical hybrid system.

which made it possible to manufacture the diffractive layer using fused filament depositioning (FDM), and second, it uses only linear operations. These two points will be further discussed and analysed in section 3.2 where a model based on this approach will be derivated.

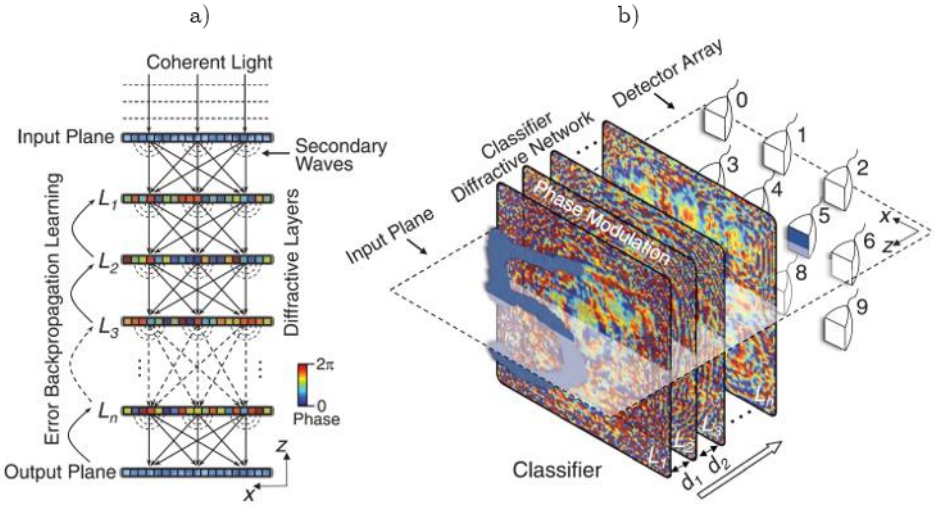


Figure 3-5: Diffractive deep neural networks (D^2NNs). a) A D^2NN comprises multiple transmissive (or reflective) layers, where each point on a given layer acts as a neuron, with a complex-valued transmission (or reflection) coefficient. The transmission or reflection coefficients of each layer can be trained by using deep learning to perform a function between the input and output planes of the network. After this learning phase, the D^2NN design is fixed; once fabricated or 3D-printed, it performs the learned function at the speed of light. b) Trained and experimentally implemented D^2NN classifier for handwritten digits and fashion products. [9]

Based on the implementation of an optical neural network above one further development must be mentioned in this context. This publication proposed a Fourier-space diffractive deep neural network [47]. Yan, Wu et al. demonstrated a convolutional front-end with nonlinear activation functions using a layer of photorefractive crystals. By modulating the wavefront phase in focal plane of a 4-F setup, a two-dimensional convolution kernel is realized. They demonstrated their design segmenting image and video data. The proposed system demonstrates an additional building block for a full passive implementation of a classification neural network. The basic setup of Yan, Wu et al. is shown in Figure 3-6

Based on the last two setups [9, 47] a mathematical model will be derived in the next chapter and the approaches will be discussed.

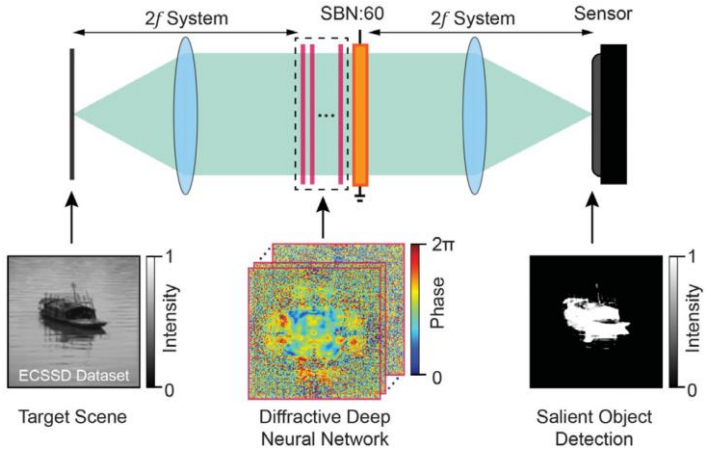


Figure 3-6: Salient object detection with Fourier-space diffractive deep neural network (F-D²NN). The F-D²NN optical image processing module is formed by inserting the D²NN along with a photorefractive crystal (SBN:60) at the Fourier plane of an optical system under coherent light. F-D²NNs can achieve all optical segmentation of the salient objects for the target scene after deep learning design of modulation layers. [47]

3.2 Deep Diffractive Neural Networks

3.2.1 Mathematical Model of an Optical Feed-Forward Network

The D²NN described in [9] uses a complex wavefront modulation to encode an amplitude and/or phase information into an optical wavefront, described in more detail later on in section 4.3. Each element of one diffractive layer acts as a neuron. Regardless of how this information is physically projected into the wavefront, one layer of the D²NN network has the following form:

$$a_j^l = f\left(\sum_k a_k^{l-1} t_j^l w_{jk}^l\right) = f\left(z_j^l\right) \tag{3-1}$$

Where a_j^l is the complex output value of neuron j at layer l and a_k^{l-1} is the complex input value from the previous layer $l-1$ of neuron k. The transmission t_j^l is a complex multiplicative bias term containing the network parameters. In [9] the phase modulation is realized by a diffractive optical element (DOE) which has a locally varying thickness with constant refractive index. The information of the propagation through space between diffractive layers is contained in w_{jk}^l connecting each neuron of adjacent layers and f is the activation function of each layer. In Figure 3-7 the notation conventions of equation (3-1) are shown.

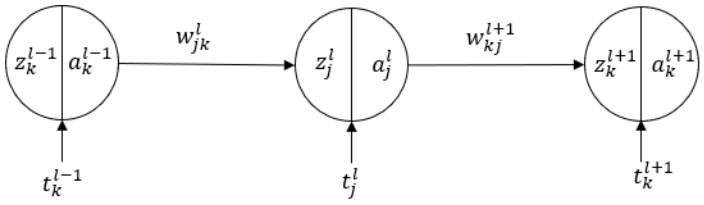


Figure 3-7: Three neurons in a diffractive deep neutral network with the indexing used in this work.

The weight matrix w_{jk}^l is determined by the physical location of each neuron k with respect to the position of neuron j i.e., the radial distance r between these to neurons. Each neuron is assumed to be a point source of a wave modulated by t_j^l The function that transforms a point source on a plane with position x and y to another point at ξ and η on a destination plane with distance Δz can be described by:

$$w_{jk}^l = \frac{1}{r} e^{ikr} \left(\frac{1}{i\lambda} + \frac{1}{2\pi r} \right) \frac{\Delta z}{r} \tag{3-2}$$

Where $r = \sqrt{(x-\xi)^2 + (y-\eta)^2 + \Delta z^2}$ is the radial distance and λ is the operating wavelength, $k = \frac{2\pi}{\lambda}$ is the wavenumber and $i = \sqrt{-1}$. See chapter 4 for the derivation of equation (3-2). The coordinate system of equation (3-2) is shown in Figure 3-8.

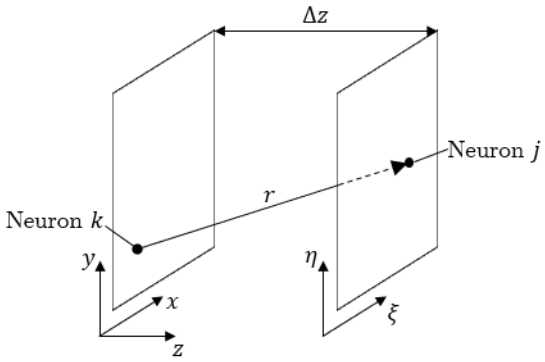


Figure 3-8: Coordinate system for two physically space diffractive neural network layers with the radial distance of a neuron k to a neuron j indicated by r.

Note that the propagation with equation (3-2) is the method used in [9], for the forward propagation another method is developed in chapter 4.

The complex transmission function of each neuron t_j^l is explained in chapter 4.3. For now, it is just defined as a phase and amplitude modulation, formally written as:

$$t_j^l = T_j \cdot e^{i\Delta\varphi_j} \tag{3-3}$$

Where T_j is the amplitude modulation and $\Delta\varphi_j$ the phase modulation term of neuron j. To reduce the indexing formalities in equation (3-1), each value can be written in a vectorized matrix notation. Hereby every two-dimensional plane is represented as a column vector. Let

$N \times M$ be the number of elements in a matrix x_{nm}, then the vectorized matrix is: $[x_{11}, x_{12}, x_{13} \dots, x_{21}, x_{22}, x_{23}, \dots, x_{NM}]^T = (x_j)^T$. Expressing the complex activation of a layer in vectorized form:

$$\boldsymbol{a}^l = f^l\left(\boldsymbol{w}^l \boldsymbol{t}^l \boldsymbol{a}^{l-1}\right) = \boldsymbol{f}^l \circ \boldsymbol{w}^l \boldsymbol{t}^l \boldsymbol{a}^{l-1} \tag{3-4}$$

Whereby $\boldsymbol{a}^{l-1} \in \mathbb{C}^1$ is the vectorized matrix of the layer before, $\boldsymbol{w}^l \in \mathbb{C}^2$ is the complex weight matrix, $\boldsymbol{t}^l \in \mathbb{C}^2$ is the diagonalized modulation of layer l. $\boldsymbol{f}^l$ is the tensor operation for applying the activation function. To further specify, $\boldsymbol{t}^l$ is structured as follows:

$$\boldsymbol{t}^l = \begin{pmatrix} t_1^l & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & t_j^l \end{pmatrix} \tag{3-5}$$

Where j is the neuron number. And the weight matrix's indices are ordered in $\boldsymbol{w}^l$ as follows:

$$\boldsymbol{w}^l = \begin{pmatrix} w_{11}^l & \cdots & w_{1k}^l \\ \vdots & \ddots & \vdots \\ w_{j1}^l & \cdots & w_{jk}^l \end{pmatrix} \tag{3-6}$$

With the equations (3-4), (3-5) and (3-6) the activation of the last layer is then:

$$\boldsymbol{a}^{L+1} = \boldsymbol{w}^{L+1} \prod_{l=L}^{1} \left(\boldsymbol{f}^l \circ \boldsymbol{w}^l \boldsymbol{t}^l\right) \cdot \boldsymbol{a}^0 \tag{3-7}$$

Note that, the product counts layers backwards form L to 1 because to the matrix multiplication is not commutative. In Figure 3-9 an illustration of the forward propagation for a three-layer D^{2}NN is shown. The transmission modulation at the input layer $l = 0$ generates the input image. The wave is modulated in each layer by the diffractive layer in that layer. In the example shown in Figure 3-9 only the phase of the wavefront is modulated.

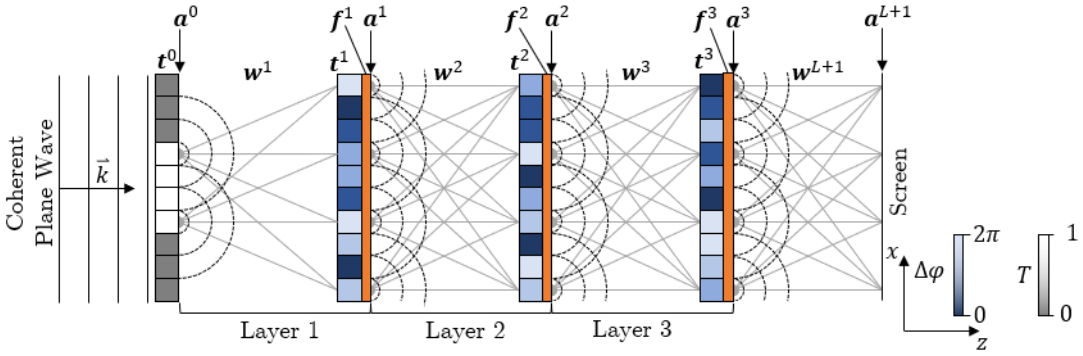


Figure 3-9: Forward Propagation of a D^{2}NN with three difractive layers according to [9] and [48]. An incoming monochromatic plane wave is modulated by $\boldsymbol{t}^0$ in amplitude to form the input image. Through several layers of propagation $\boldsymbol{w}^l$, modulation $\boldsymbol{t}^l$ and activation $\boldsymbol{f}^l$ the diffracted optical wave $\boldsymbol{a}^{L+1}$ falls onto an observation screen.

The activation function $\boldsymbol{f}^l$ is a function of the modulated input. If L is the number of diffractive layers, then the screen where the optical field is evaluated is at $L+1$.

At the screen, detectors are located. The detectors measure the intensity of the optical wave. The intensity at the detector screen is the absolute squared of the optical field[20] i.e., the activation function $\boldsymbol{a}^{L+1}$ at the last layer $L+1$. The intensity vector $\boldsymbol{I}$ then is:

$$\boldsymbol{I} = |\boldsymbol{a}^{L+1}|^2 = \left| \boldsymbol{w}^{L+1} \prod_{l=N}^{N} \left(\boldsymbol{f}^l \boldsymbol{w}^l \boldsymbol{t}^l \right) \cdot \boldsymbol{a}^0 \right|^2 = \boldsymbol{a}^{L+1} \circ (\boldsymbol{a}^{L+1})^* \tag{3-8}$$

Where $(\boldsymbol{a}^{L+1})^*$ is the complex conjugate of $\boldsymbol{a}^{L+1}$. In the output plane, specific regions are assigned to specific output classes. The resulting relative optical intensity per classification region resembles the activation of the classification layer. An example of a classification plane, as used in [9] for $n = 10$ classes, is shown in Figure 3-10. The classification is therefore done digitally by evaluating a camera image or electronically by measuring the output of an optical detector that might be multiple photodiodes or a camera.

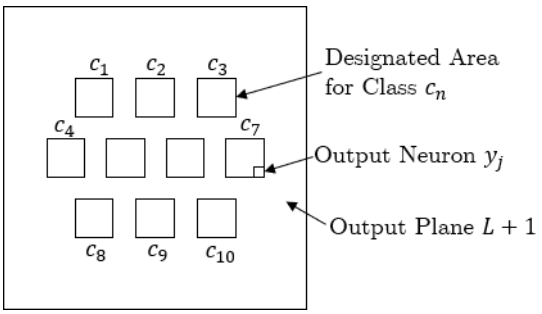


Figure 3-10: Classification Layer of a D²NN at the layer $L+1$ of the network with each image point resampling an output neuron y_j. Every output neuron that falls into the region c_n is assigned to the class number n.

3.2.2 Nonlinearity in D²NNs

The network architecture proposed by Lin, Riverson et al. [9] has a inherent flaw, as the researches noted in the supplementary material of [9], that is the abscence of nonlinear activations inside the network. Calling it a „deep“ neural network therefore only can be made assuming nonlinearity can be implemented. Althought the D²NN presented was comprised of five diffractive layers, mathematically those linear oprations done by the diffractive layers should be also possible using one layer. Despite of this fact, it was in a later analysis [46] shown that a diffractive neural network benefits from additional layers. Although all materials are

[20] See chapter 4.1.1 equation (4-17) for more details.

basically nonlinear, this can hardly be the reason for the increasing accuracies reported. Mengu, Luo et al. [46] hypothesize that, an increase in diffracting neuron coupling and diffraction efficiency cause the network to perform better if more layer are present. Until now, no complete argument has been made what exactly gives a linear D^2NN its computational depth.

Regardless, a D^2NN should improve using nonlinear activation functions [9, 46]. Therefore, Zhou et al. [48] and Yan, Wu et al. [47] used a ferroelectric thin-film of $1\,mm$ thickness (SBN:60, Strontium barium niobate [49]) as nonlinear medium applying a nonlinear activation function. [45] stated the phase shift of the ferroelectric thin-film to be: $\Delta\varphi(E) = \pi \cdot \frac{|E|^2}{1+|E|^2}$ with $|E|^2$ as the local intensity, without any further context. As this research [45, 47] was published last year (2019), further advancements will be hopefully made in the near future. But this shows that nonlinearities can be implemented by using SBN:60 nonlinear crytsals, and increase the capability of D^2NNs to solve even more complex tasks.

3.2.3 Diffractive vs. Classical Neural Networks

The mathematical difference of the D^2NN concept with respect to classical NNs described in chapter 2 is the use of complex values. Although it has been proposed in the mid-20th century [50] most basic networks use real values. In optical applications it is inevitable to use complex notation for describing at least the basic properties of an electro-magnetic wave. By using only one complex number the polarization and magnetic interaction are still omitted. Nevertheless, the use of complex values is in no sense a disadvantage and has been used in specific neural networking tasks [51].

A further difference is the use of a multiplicative bias term, in form of a complex transmission function. The classical NN uses an additive bias term. The use of the term “bias”, which stems from preloading a neuron, therefore becomes only a remnant in case of D^2NNs. The weights, although not fixed values, are determined by the spatial locations of the neurons with respect to each other and are not trainable parameters in this setup. The complex transmission on the other hand has two values which both contribute to one complex modulation value, that are the amplitude transmission and the phase retardation.

The maximum size of classical NNs is given by the computational power needed for training. In contrast, the size of D^2NNs is limited by a needed signal-to-noise ratio (SNR) of the detector at the output layer and by the manufacturing accuracy of the diffractive layers. The main loss of SNR at the detector is the modulation of light amplitude in diffractive layers [9]. When

implementing phase-only diffractive layers, this loss can be mitigated to zero. The remaining losses are absorption and scattering inside the diffractive layer material, absorption and scattering between layers[21] and energy coupling into higher diffractive orders. The latter has two impacts which must be differentiated. For this purpose, a most basic DOE, a binary phase grating, is analyzed. A binary grating is described by its grating constant g which is the reciprocal of distance d between each grating period $g = \frac{1}{d}$. When coherent monochromatic light with the wavelength λ hits the grating at an inclination angle of $0°$ to the grating surface normal, the resulting angles φ_n at which diffraction maxima occur can be described by the following equation:

$$\varphi_n = \sin^{-1}(n\lambda g) \tag{3-9}$$

Where n is the integer number of the diffraction order. An illustration of this basic principle is shown in Figure 3-11.

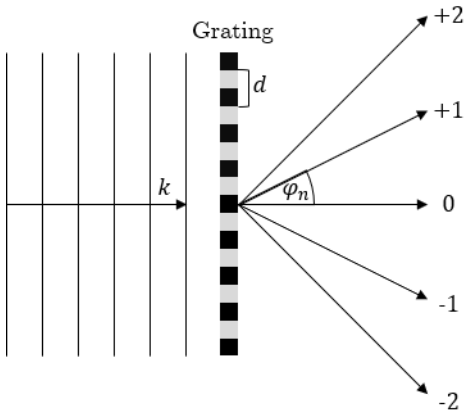


Figure 3-11: Basic binary optical grating, where the grating has either a changing amplitude from 1 to 0 or a changing phase from 0 to π.

From equation (3-9) it becomes clear that smaller spacings between grating periods result in higher diffraction angles. Smaller spacings of structural differences correspond to higher spatial frequencies of the DOE structure. If now a discrete surface is calculated, the smallest feature size result in diffractive orders according to (3-9). When a DOE is used to generate a holographic image, multiples of those images exist at angles which correspond to the discretization distance. In case of the FDM printed DOE of [9], this distance d would be the line distance between two printed lines, commonly $0.4\,mm$.

[21] The scattering and absorption between layers refer for example to particles scattering the wave as it propagates. These effects can be mitigated encapsulation or operation in vacuum.

If the surface is perfectly continuous, or with at least sampled at a distance of λ, representing the ideal surface, then every diffraction order except the 0^{th} order would be at $\varphi_{n\neq 0} \geq 90°$. Effectively this means, that all energy of the incoming wave is within the diffraction image in the centre. Otherwise, multiple images at higher angles take energy from the 0^{th} order image.

A typical example of a discretized surface is shown in Figure 3-12. Hereby, a sinusoidal pattern with the spatial frequency $f_1 = d_1^{-1}$ is sampled by rectangular steps with the periodicity $f_2 = d_2^{-1}$. The resulting diffraction pattern would have the lower frequency diffraction pattern of the sinusoidal pattern but also an infinite number of multiple images du to the rectangular pattern, the first according to f_2 the next du to integers multiple of f_2.

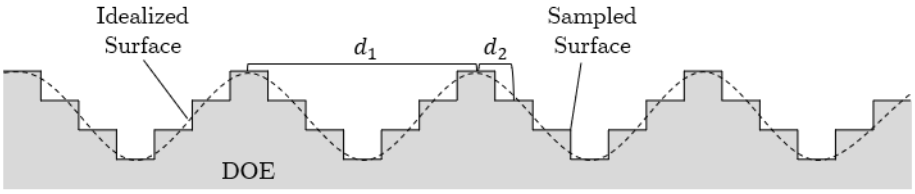


Figure 3-12: Sketch of two discrete spatial frequencies resulting from a sampled surface.

One effect of this imperfection is a lower SNR at the detector due to energy in higher diffraction orders. A common measure of this is called diffraction efficiency. The other effect would be a given minimum distance between diffractive layers of the D^2NN and a minimum neuron feature size, so that network diffraction is distinguishable from discretization diffraction. Or for a variable DOE size a maximum neuron density per area.

Overlapping multiple images (aliases) causes crosstalk. This crosstalk effect is shown in Figure 3-13 a). A DOE with a higher minimum feature size is shown in Figure 3-13 b).

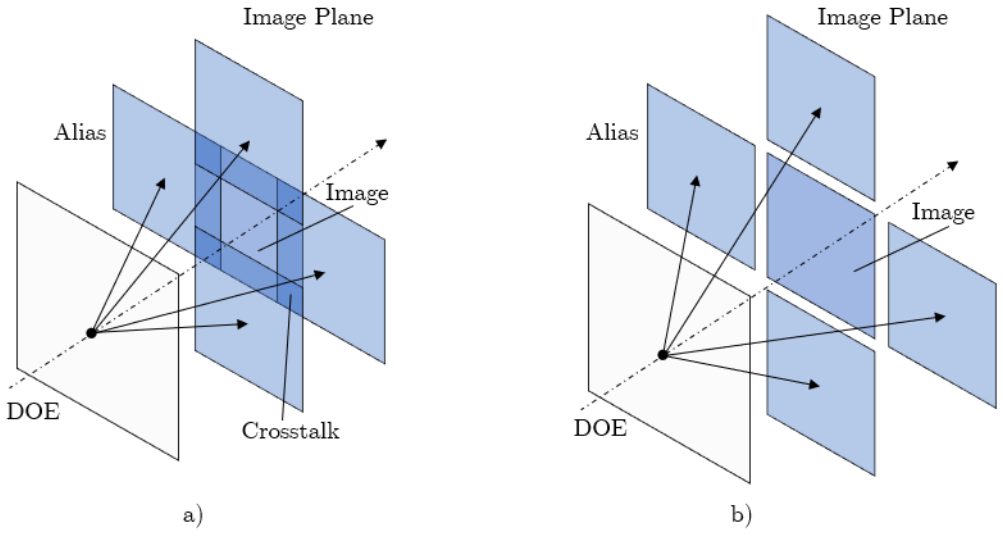


Figure 3-13: Multiple diffraction images due to surface discretization with crosstalk in the overlap region a) and with no crosstalk in b).

There will always be to some extend higher frequency disturbance due to manufacturing and calculation discretization.

However, the number of neuron density has also a practical limit, that is defined by manufacturing limitations. Besides those three points, D^2NNs are a quite similar physical resemblance of digital neural networks, as the backpropagation algorithm for optical networks points out as well.

3.2.4 Backpropagation in Deep Diffractive Neural Networks

The backpropagation algorithm for D^2NNs has been described by Lin et al. [52] Hughes, Minkov et al. [53] and Zou, Li et al. [45] on the basis of complex domain backpropagation [17]. According to [45], the derivative of the loss function L in respect to a phase modulation $\Delta\boldsymbol{\varphi}^l$ at layer l can be formulated as:

$$\frac{\partial L}{\partial \boldsymbol{\Delta\varphi}^l} = \frac{\partial L}{\partial \boldsymbol{a}^{L+1}}\frac{\partial \boldsymbol{a}^{L+1}}{\partial \boldsymbol{\Delta\varphi}^l} + \frac{\partial L}{\partial (\boldsymbol{a}^{L+1})^*}\frac{\partial (\boldsymbol{a}^{L+1})^*}{\partial \boldsymbol{\Delta\varphi}^l} = 2\mathcal{R}e\left\{\left(\frac{\partial L}{\partial I} \circ (\boldsymbol{a}^{L+1})^*\right)^T \frac{\partial \boldsymbol{a}^{L+1}}{\partial \boldsymbol{\Delta\varphi}^l}\right\} \tag{3-10}$$

Whereby $\boldsymbol{\Delta\varphi}^l$ is a phase modulation matrix of layer l, defined in $t_j^l = T_j \cdot e^{i\Delta\varphi_j}$ (3-3), L is the loss function and $\boldsymbol{a}^{L+1}$ the matrix of the optical field at the detector plane. The *- notation means the complex conjugate and $\mathcal{R}e$ denotes the real part of a complex number. Note that, if $z \in \mathbb{C}$ then $z + z^* = 2\mathcal{R}e\{z\}$ and $\frac{\partial f(z)}{\partial z} = \frac{\partial f(z)}{\partial z} + \frac{\partial f^*(z)}{\partial z^*}$. The error field $\boldsymbol{\delta}^{L+1}$ at layer $L+1$ may be defined as:

$$\boldsymbol{\delta}^{L+1} = \frac{\partial L}{\partial I} \circ (\boldsymbol{a}^{L+1})^* \tag{3-11}$$

With the definition of $\boldsymbol{\delta}^{L+1}$, equation (3-10) can be simplified to:

$$\frac{\partial L}{\partial \boldsymbol{\Delta\varphi}^l} = 2\mathcal{R}e\left\{(\boldsymbol{\delta}^{L+1})^T \frac{\partial \boldsymbol{a}^{L+1}}{\partial \boldsymbol{\Delta\varphi}^l}\right\} \tag{3-12}$$

The derivative of the optical field with respect to phase modulations of each layer is, according to [45]:

$$\frac{\partial \boldsymbol{a}^{L+1}}{\partial \boldsymbol{\Delta\varphi}^l} = \begin{pmatrix} i \cdot diag\left(\boldsymbol{a}^l \boldsymbol{w}^l \prod_{k=l-1}^{1} \left(\boldsymbol{f}^k \circ \boldsymbol{a}^k \boldsymbol{w}^k\right) \boldsymbol{a}^0\right)^T \cdot \\ diag\left(\boldsymbol{f}^{l\prime}\right) \prod_{k=l+1}^{L} \left(\left(\boldsymbol{w}^k\right)^T \boldsymbol{a}^k diag\left(\boldsymbol{f}^{k\prime}\right)\right) (\boldsymbol{w}^{L+1})^T \end{pmatrix}^T \tag{3-13}$$

Where $\boldsymbol{f}^{l'}$ is the derivative of the activation function of each layer and $diag()$ represents the diagonal matrix of a vector, so that $diag(\boldsymbol{z}) = \begin{pmatrix} z_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & z_{jj} \end{pmatrix}$. Also note that $diag(\boldsymbol{z}) \cdot \boldsymbol{w} \equiv \boldsymbol{z} \circ \boldsymbol{w}$. Zuo, Li et al. [45] showed that, when inserting equation (3-13) into equation (3-12), the result can be formulated as:

$$\frac{\partial L}{\partial \boldsymbol{\Delta\varphi}^l} = 2\mathcal{R}e\{i \cdot \boldsymbol{p}_f^l \circ \boldsymbol{p}_b^l\} \tag{3-14}$$

Whereby:

$$\begin{aligned} \boldsymbol{p}_f^l &= \boldsymbol{a}^l \boldsymbol{w}^l \prod_{k=l-1}^{1} \left(\boldsymbol{f}^k \circ \boldsymbol{a}^k \boldsymbol{w}^k\right) \boldsymbol{a}^0 \\ \boldsymbol{p}_b^l &= diag\left(\boldsymbol{f}^{l'}\right) \prod_{k=l+1}^{L} \left(\left(\boldsymbol{w}^k\right)^T diag\left(\boldsymbol{f}^{k'}\right) \boldsymbol{a}^k\right) \left(\boldsymbol{w}^{L+1}\right)^T \cdot \boldsymbol{\delta}^{L+1} \end{aligned} \tag{3-15}$$

Here, both matrices $\boldsymbol{p}_f^l$ and $\boldsymbol{p}_b^l$ can be interpreted as a forward propagating optical field and a backpropagating error field, respectively. The transposed weighting matrix $\left(\boldsymbol{w}^k\right)^T$ represents the propagation of the backward travelling error field. Note that the matrices $\boldsymbol{a}^l$contain the modulation by the diffractive layers $\boldsymbol{t}^l$ according to the definition in (3-4). As in subsection 2.2.2 explained, all phase gradients $\frac{\partial L}{\partial \boldsymbol{\Delta\varphi}^l}$ are averaged over all samples in one batch, so that:

$$\Delta\boldsymbol{\varphi}^l = \frac{1}{N} \sum_{n=1}^{N} -\eta \frac{\partial L_n}{\partial \boldsymbol{\Delta\varphi}_n^l} \tag{3-16}$$

With the averaged phase gradients, the individual phase modulations of the diffractive layers are updated. In contrast to the forward propagation, in the error calculation and backpropagation it is not necessary to do more precise calculation, because the error propagation is not limited by the sampling theorem of the Rayleigh-Sommerfeld weight matrix $\boldsymbol{w}^l$. So, for example, when calculating the backpropagation of a 200×200 neurons per layer network the backpropagation weight matrix only needs to be the size of $200^2 \times 200^2$ independent of the actual size of each neuron. For the forward propagating field $\boldsymbol{p}_f^l$, a downsampled version might be used, that takes the mean value of the optical field over an area corresponding to a neuron.

3.2.5 Convolutional Deep Diffractive Neural Networks

A natural step for D²NNs is, due to its two-dimensional property, to solve tasks in image processing and classification. As explained in section 2.4, a certain degree of generalization of shift variant data is needed for image classification. That means for example recognizing the number five in an image wether it is located in the top left corner of the image or somewhere else. To achieve this convolutional and pooling layers are a good option, as established in section 2.4. Especially in optics convolutional calculations are straight forward to implement. The basic concept is hereby, that when an optical wave is focused by a lens[22], the image in the focal plane is the spatial spectrum of the focused wave [54, pp 118, 976-977]. Modulations in the spectrum of an image are convolutions with the modulation as convolution kernel[23] [32, pp 220–222]. This property can be exploited making the kernel i.e., aperture transmission function, a trainable parameter in the D²NN training. The first published proposal by Yan, Wu et al. proved the concept by applying a one-kernel D²NN to an image segmentation and a classification tasks, which they then called a Fourier-space diffractive deep neural network (F-D²NN). As the transmission function in real space, the transmission function in spatial frequency space is:

$$\boldsymbol{t}^l = \boldsymbol{T}^l \cdot e^{i\Delta\boldsymbol{\varphi}^l}$$

With the transmission function $\boldsymbol{t}^l$, the optical field at the convolution layer output $\boldsymbol{a}^l$ becomes:

$$\boldsymbol{a}^l = \boldsymbol{a}^{l-1} * \boldsymbol{t}^l$$

A thin plate put in the focal plane of a 4f-system acts now in the spatial spectrum of the object. This allows the system to detect spatial dependent features.

A convolutional unit with one kernel is shown in Figure 3-14 a). Convolution units can be sequenced to increase network accuracy. As [47] showed, a convolutional diffractive network profits from nonlinear activations that may be put between convolution units, where the image plane in Figure 3-14 a) is. An example of an edge filter is shown in Figure 3-14 b) using a Fourier transform and a high pass spatial filter. The cut-off frequency of the filter is shown by a black circle in the spatial frequency image. The filtered image is then backtransformed into the spatial domain.

[22] Focusing might be done using diffractive optical elements too.
[23] See subsections 4.1.5 and 4.2.1 for more details on transformations and convolution.

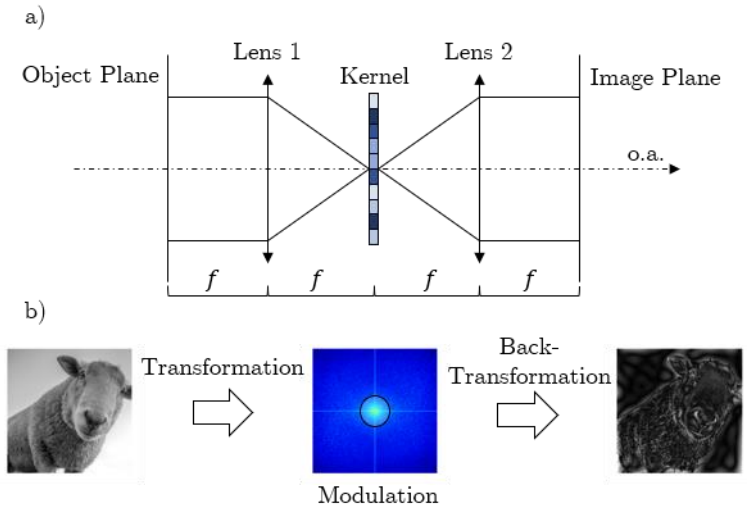


Figure 3-14: Illustration of a convolutional unit in an F-D²NN; a) shows the optical setup with two lenses at a modulating kernel in the focal (Fourier) plane; b) example of an image modulated in the Fourier plane with a high-pass filter.

3.3 Hyphotheses & Further Developments of F-/D²NNs

The D²NN framework presented by [9] uses an approach which treats every neuron of a diffractive layer as a source of one secondary elemental point source and calculates one point in the input plane to the next layer as a superposition of all point sources of the previous layer. Lin, Riverson et al. prooved in [9] that this approach is sufficient for neuron sizes smaller than the operation wavelength by achieving a high prediction accuracy of their network. This was possible due to the use of terahertz wavelength of $\lambda = 1\,mm$ in air ($0.3\,THz$) as operating wavelength. The material used in [9] was Polyactic acid (PLA) which has a refractive index of around 1.9 [55] at $0.3\,THz$. Typical FDM printers are able to print with a height resolution of $0.1\,mm$. The optical path difference (OPD) achievable with this setup becomes: $OPD = \Delta n \Delta d = 0.09\,mm = 0.09\,\lambda$. With a typical lateral resolution of $0.4\,mm$ a neuron would have the size of $0.4\,\lambda$, as Lin, Riverson et al. used a layer size of $8 \times 8\,cm^2$ with a neuron count of 200×200 and $\frac{80\,mm}{200} = 0.4$.

To compare those dimensions when transitioning from tera hertz to near-infrared (NIR) and visibile (VIS) wavelengths the example above, based on the neuron count and absolute number of neurons per layer, is recalculated. For example, when using a wavelength of $\lambda = 632.8\,nm$ and a typical refractive index change of $\Delta n = 0.5$ the needed height resoultion of a diffractive layer would become $\Delta d = \frac{0.09\,\lambda}{0.5} \approx 114\,nm$. The lateral neuron size of $0.4 \cdot 632.8\,nm =$

$253.12\,nm$. Although both requirements of lateral and height resolution are physically possible to realize, they do require highly sophisticated equipment.

Further, this setup results in a diffractive layer size of $0.05 \times 0.05\,mm^2$, which is way too small to fit on any optical sensor with the necessary spatial resolution or would need additional optical elements for an appropriate magnification. When increasing the neuron count per layer drastically, of course a larger diffractive layer might be manufacturable, but this larger neuron count is not beneficial for networks' convergence and accuracy. A too large neuron count causes an effect which is called overfitting [10, chapter 3].

Keeping the neuron count per layer constant but increasing the neuron size to be much larger than the wavelength, would result in a large number of wavelets per neuron necessary, because of the Rayleigh-Sommerfeld calculation used in [9]. The sampling condition for the Rayleigh-Sommerfeld integral has been investigated by Mehrabkhani, Schneider [56] and is:

$$\Delta s \leq \frac{\lambda\sqrt{a^2 + \Delta z^2}}{2\Delta z} \tag{3-17}$$

Where Δs is the sampling distance, λ is the wavelength, a is the aperture diameter and Δz is the distance between two calculation planes. This distance in the simplest case would be the distance between the diffractive layer and the detector. Clearly the sampling interval depends on the distance Δz if the aperture diameter is given. Therefore, Figure 3-15 shows the needed sampling interval depending on the aperture size, for different distances Δz. The lowest needed resolution independent of the distance Δz is $\frac{\lambda}{2}$. If now a weight matrix connecting every sampling point of each plane with every other sampling point, the resulting matrix would have the size of $N_{weights} = N_x^4 = \left(\frac{a}{\Delta s}\right)^4$ [24]. This weight matrix size i.e., the number of calculations necessary is shown in Figure 3-15 as dashed line for each distance Δz. The point of this example is, that when a diffractive layer is scaled to sizes that fit on NIR and VIS detectors, a.k.a. camera chips, the processing power needed to calculate every network layer for every training example becomes so large that the time needed for network training is not feasible anymore.

[24]Assuming a squared aperture enclosing the aperture with the diameter a, so that the edge length is a.

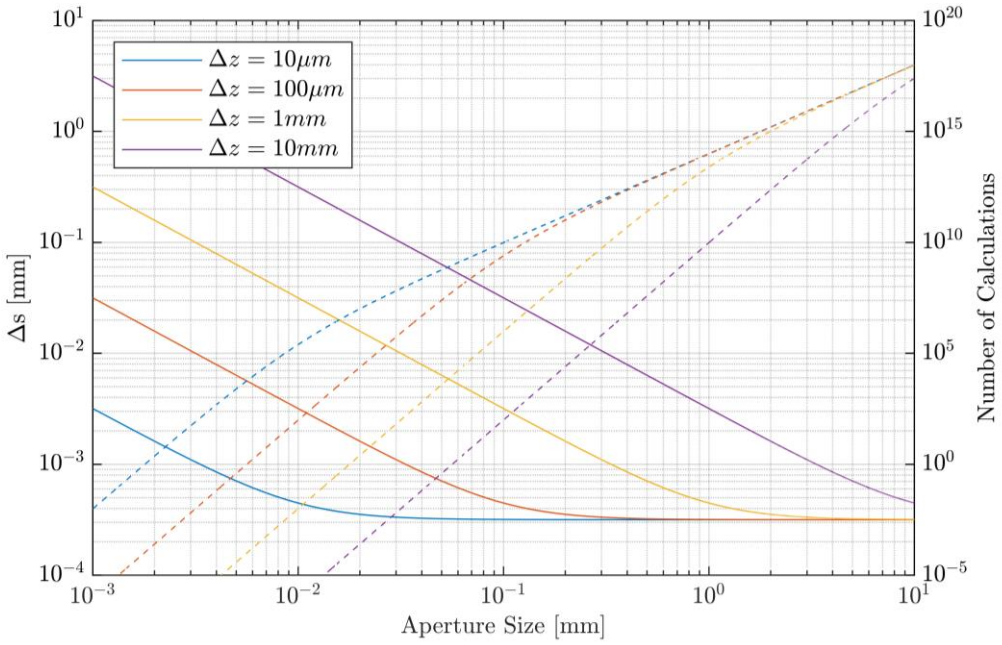


Figure 3-15: Minimum sampling distance Δs for the Rayleigh-Sommerfeld integral for several distances Δz as a function of the aperture size a shown as continous line and the corresponding number for weight multiplications or calculations per diffraction layer is shown as dashed lines.

Based on the massive computational load of the Rayleigh-Sommerfeld diffraction calculation, the first hypothesis of this thesis is:

Hypothesis 1

With a more efficient algorithm to calculate the diffracted optical field, the forward propagation for training a D²NN for NIR and VIS operation wavelength, can be calculated in a macroscopic scale on a common desktop PC.

A supplementary hypothesis to the first one above comes from the fact that manufacturing of small structures always comes with a certain degree of inaccuracy. Shi, Chen et al. [57] already prooved that by injecting white phase noise into the training process, the perfomance of the network improved. With those results it can be conculded that by modeling surface deviations into complex transmission coefficients the robustness of the actual accuracy can be increased. So, an addition to the first hypothesis is:

Hypothesis 2

The algorithm of Hypothesis 1 is fast enough to allow subsampling of neuron structures to include surface deviations into the training of a D²NN and still remain reasonably fast when using NIR or VIS wavelengths.

A third motivational goal relates to the F-D²NN framework shown in subsection 3.2.5. Hereby, Yan, Wu et al. [47] made a proof of concept for a holistic image filter emphasizing on salient objects in an image scene. According to the digital role model, a CNN [30] front-end combined with a feed-forward network has massive advantages regarding image recognition. This is due to the fact that the multiple convolutions in each layer work out distinct features in the image. Those features are condensed to a small number or even one neuron value per feature. This set of weighted feauture can be easily processed and classified by a feed-orward network. The proposal of Yan, Wu et al. Lacks this property, as it only acts as a filtering front-end that benfits one type of feature. Therefore, a setup is proposed here, as the one shown in Figure 3-16. The idea is that a grating splits an image into multiple copies. Those copies are collimated by a lens and then focused on a diffractive layer in the Fourier plane. The diffractive layer modulates amplitude and phase of each copy separatly. A lens array then collimates each convolution path. When concatenating more of the convolutional cells, shown in Figure 3-16, a set of features abstracted from an image can be passed to an adjacent feed-forward network. Theoretically, this concept allows true feature based detection, since multiple specific features are processed by a D²NN and not one spatial distriubution of one feature, as it is the case in [47].

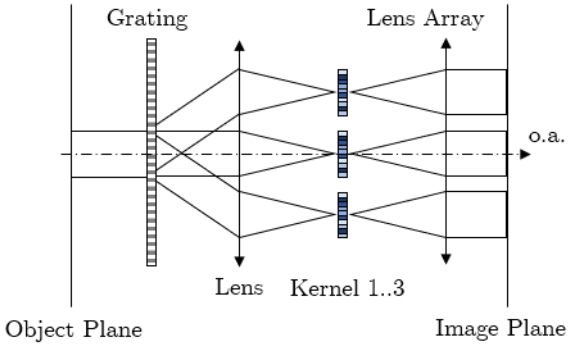


Figure 3-16: Concept of a multi-kernel convolution layer unit based on a diffraction grating, a lens, multiple diffractive kernels, and a lens array.

Based on the concept of Figure 3-16, the third hypothesis of this thesis is:

Hypothesis 3

A physically accurate forward propagation model can reproduce a multiple kernel convolution in one computational task, according to the concept of Figure 3-16, for the use of a Fourier-space convolutional deep diffractive neural network.

The concept of Hypothesis 3 might be realized using only separate convolutions and stitch all convolutional fields together before passing it to the feed-forward network. Using an actual simulation for the forward pass, the crosstalk between Fourier-space images as well as misalignment errors might be included in further works on this topic. Also, the multi-kernel convolution unit might serve as an example for the computational power of the algorithm derived for Hypothesis 1 and Hypothesis 2. A suitable way for calculating a diffractive pattern for the forward propagation in a D^{2}NN is the topic of the next chapter. Starting from Maxwells equations, the angular spectrum method is derived, and the concept of a band-limited angular spectrum method is introduced to meet the expectations of Hypothesis 1 and Hypothesis 2.

4 Numerical Scalar Diffraction Simulation

4.1 Analytical description of scalar diffraction

To model the forward propagation between two diffractive neural network layers described in section 3.2 and 3.3, a theory of diffraction and a numerical implementation will be examined in this chapter. The goal is to develop a theoretical approach to model the modulation of each network layer i.e., a diffractive layer and a resulting electrical field distribution at an output plane. The diffraction of light refers to the observable phenomena of light passing through a modulating aperture, travelling through space, and interfering at a detector. The intensity distribution at the observation screen is called a diffraction pattern. When one would try to describe diffraction using only ray optics, the observed intensity distribution would only be the geometrical shadow of the aperture, illuminated by a light source. Sommerfeld introduced the term diffraction by:

> *"...any deviation of light rays from rectilinear path which cannot be interpreted as reflection or refraction". [Sommerfeld 1894]*

For describing the effects of diffraction, a model based on wave optics is required. A comparison is shown in Figure 4-1, whereby a geometrical model a) may be a sufficient model for apertures much larger than the wavelength of the source. Figure 4-1 b) illustrates a more precise view on the nature of light, where the wave characteristics of light are considered.

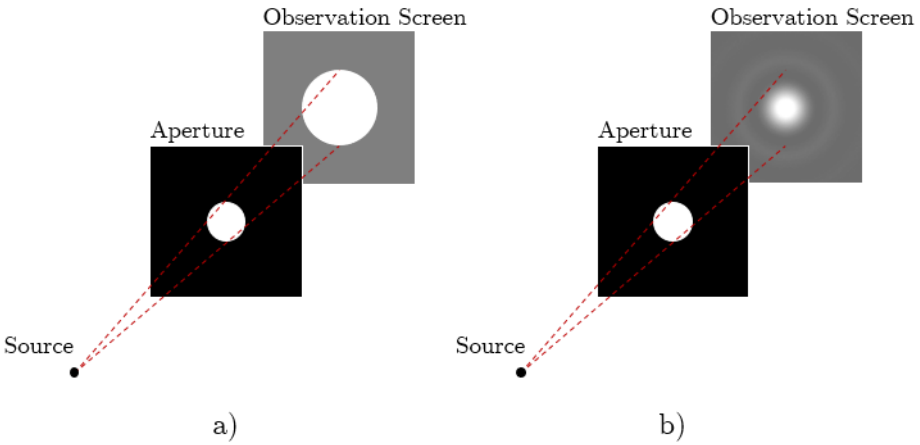


Figure 4-1: Imaging of a point source through an aperture, where a) only takes geometrical ray optics into account and b) shows an intensity pattern according to wave optics.

4.1.1 From the Maxwell Equations to the Scalar Wave Equation

A description of waves is derived from the Maxwell equations (4-1) to (4-4), which give a fundamental description of the relation between electric and magnetic fields. Furthermore, equations (4-1) to (4-4) predict that transversal waves travel at the speed of light, to conclude that light might be viewed as an electromagnetic wave. Maxwell's equation consist of Gauss' law of electricity (4-1), Gauss' law of magnetism (4-2), Faraday's law of induction (4-3) and Ampere's law with Maxwell's addition of the displacement current density $\boldsymbol{J}$ (4-4):

$$\nabla \cdot \boldsymbol{D} = \rho \tag{4-1}$$

$$\nabla \cdot \boldsymbol{B} = 0 \tag{4-2}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{4-3}$$

$$\nabla \times \boldsymbol{H} = \frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{J} \tag{4-4}$$

Where the differential nabla operator is $\nabla \equiv \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$, $\boldsymbol{D} = \varepsilon \boldsymbol{E}$ is the electric density or electric displacement, $\boldsymbol{E}$ is the electrical field, ε is the absolute permittivity, ρ is the electric charge density, $\boldsymbol{B} = \mu \boldsymbol{H}$ is the magnetic flux density, μ is the absolute permeability and $\boldsymbol{H}$ is the magnetic field strength. Taking the curl of (4-3) results in:

$$\nabla \times \nabla \times \boldsymbol{E} = -\mu \frac{\partial}{\partial t}(\nabla \times \boldsymbol{H}) \tag{4-5}$$

Let $\boldsymbol{A}$ be an arbitrary vector then $\nabla \times \nabla \times \boldsymbol{A} = \nabla(\nabla \cdot \boldsymbol{A}) - \nabla^2 \boldsymbol{A}$ and $\nabla(\nabla \cdot \boldsymbol{A}) = 0$ is the vector identity. If the electric charge density ρ and the current density $\boldsymbol{J}$ is zero, i.e. a source free region a.k.a. vacuum is assumed then the vector identity can be applied to equation (4-5) and inserting (4-4) leads to:

$$\nabla^2 \boldsymbol{E} = \mu \frac{\partial}{\partial t}(\nabla \times \boldsymbol{H}) = \mu \frac{\partial}{\partial t}\left(\frac{\partial}{\partial t} \boldsymbol{D}\right) \tag{4-6}$$

One can write the vectorial wave equation for the electrical field in its common form by substituting the electrical charge density and assuming a dielectric, linear, isotropic, homogenous, nondispersive, and nonmagnetic medium:

$$\boxed{\nabla^2 \boldsymbol{E} = \mu \varepsilon \frac{\partial^2}{\partial t^2} \boldsymbol{E}} \tag{4-7}$$

Any function satisfying equation (4-7) represents an optical wave. Because the differential wave equation is linear, the principle of superposition applies [54, pp 40–41]. If one defines the real functions $E_1(\vec{r},t)$ and $E_2(\vec{r},t)$ of position $\vec{r} = \sqrt{x^2+y^2+z^2}$ as possible optical waves, then $E(\vec{r},t) = E_1(\vec{r},t) + E_2(\vec{r},t)$ also represents a possible optical wave. By reducing (4-7) by one dimension (in this case y) and choosing the z direction as the propagation direction, one arrives at the wave equation for the electrical field in x-direction propagating in the z-direction:

$$\frac{\partial^2}{\partial z^2}E_x = \mu\varepsilon\frac{\partial^2}{\partial t^2}E_x \tag{4-8}$$

An important observation is that any function in the form of $f(z \pm c \cdot t)$ will satisfy (4-8). For completeness: Any function that has well behaved derivatives will satisfy (4-8). If $E_x = f(z - c \cdot t)$, then $\frac{\partial}{\partial z}E_x = f'(z \pm c \cdot t)$ therefor $\frac{\partial^2}{\partial z^2}E_x = f''(z \pm c \cdot t)$ and $\frac{\partial}{\partial t}E_x = -c \cdot f'(z \pm c \cdot t)$ therefore $\frac{\partial^2}{\partial t^2}E_x = c^2 \cdot f''(z \pm c \cdot t)$. It is now obvious that $\frac{\partial^2}{\partial z^2}E_x = \frac{\partial^2}{\partial t^2}E_x/c^2$. Therefor the speed of light c can be defined as:

$$c = 1/\sqrt{\mu\varepsilon} \tag{4-9}$$

By using the magnetic constant $\mu_0 = 4\pi \cdot 10^{-7}\frac{H}{m}$ as the permeability of free space and the electric constant $\varepsilon_0 = 8.85418782 \cdot 10^{-12}\frac{F}{m}$ as the permittivity of free space, the speed of light in vacuum becomes:

$$c_0 = 1/\sqrt{\mu_0\varepsilon_0} \tag{4-10}$$

To define the speed of light in a homogeneous medium, the index of refraction n is defined as the ratio between the speed of light in vacuum and the phase velocity of a light wave v_p travelling through a medium with the refractive index n:

$$n = \frac{c_0}{v_p} = \sqrt{\frac{\mu\varepsilon}{\mu_0\varepsilon_0}} = \sqrt{\mu_r\varepsilon_r} \tag{4-11}$$

Where $\mu = \mu_r\mu_0$ is the absolute permeability and $\varepsilon = \varepsilon_r\varepsilon_0$ is the absolute permittivity of a medium and μ_r and ε_r are material constants.

4.1.2 The Helmholtz Equation

With the basis in form of the wave equation established in the previous section, the fundamental function of diffraction theory is presented here. A monochromatic wave, that satisfies the wave equation (4-7) [54, pp 42–43] is represented by the complex function:

$$E(\vec{r},t) = E(\vec{r})\cos\big(2\pi\nu t - \varphi(\vec{r})\big) = \mathcal{R}e\big\{E(\vec{r})e^{i2\pi\nu t}\big\} \tag{4-12}$$

Where $E(\vec{r})$ is the complex amplitude and may be represented as a phasor with a fixed amplitude A and phase φ, as shown in Figure 4-2 b). ν in Figure 4-2 a) is the optical frequency with which the phasor rotates. $Re\{\,\}$ and $Im\{\,\}$ signify the real part and imaginary part of the complex function respectively, in which $i = \sqrt{-1}$.

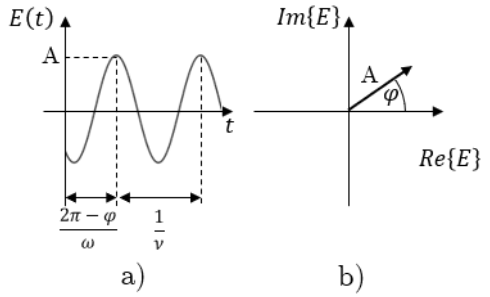


Figure 4-2: Monochromatic wave at a fixed point represented as a) a harmonic wave with frequency ν and b) as complex amplitude phasor with the vector length A and phase φ.

If (4-12) is substituted into the wave equation (4-7) the result becomes the differential equation called the Helmholtz equation [32, pp 38–39, 54, pp 43]:

$$\boxed{0 = \nabla^2 E(\vec{r}) + k^2 \cdot E(\vec{r})} \tag{4-13}$$

$k = nk_0 = \frac{2\pi\nu}{c_0} = \frac{\omega}{c_0} = \frac{2\pi}{\lambda_0}$ is the angular wavenumber. The wavenumber is the magnitude of the wavevector $\vec{k} = \sqrt{k_x^2 + k_y^2 + k_z^2}$ [54, pp 43]. In case of linear optics, the wavevector is real-valued and represents the propagation direction in three-dimensional space $\mathbb{R}^3$. When only looking at monochromatic continuous waves, equation (4-12) can be viewed at $t = 0$ and all necessary information falls exclusively on the complex amplitude $E(\vec{r})$, because $e^0 = 1$. This step effectively results in a time and space separation. So, solutions that satisfy the Helmholtz equation (4-13) in a homogenous medium are a sufficient way to describe the characteristic of the E-field of an optical wave, omitting polarization and dispersion properties. Two simple solutions are plane waves and spherical waves shown in two-dimensional space $\mathbb{R}^2$ in Figure 4-3 a) and b) respectively. The plane wave in Figure 4-3 a) has a vectorial wavevector $\boldsymbol{k}$ (in

this example with only one component $k_z \neq 0$), where as the spherical wave of Figure 4-3 b) is described by the wavenumber k only (because $k_x = k_y = k_z$).

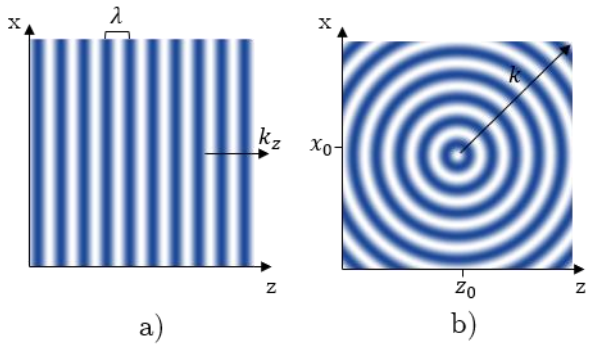


Figure 4-3: Solutions to the Helmholtz equation; a) a plane wave travelling in z-direction with the spatial periodicity λ and b) a spherical wave at the origin z_0, x_0 and $k_x = k_z$.

Formally the complex amplitude of a plane wave can be described as:

$$E(\vec{r}) = A \cdot e^{-i\boldsymbol{kr}} = A \cdot e^{-i(k_x x + k_y y + k_z z)}, \tag{4-14}$$

and the complex amplitude of a spherical wave as:

$$E(r) = \frac{A}{r} \cdot e^{ikr} \tag{4-15}$$

The description of the plane wave in equation (4-14) suggests a constant intensity of the E-field everywhere in space, because the intensity of a monochromatic wave for a constant point in time is defined by [58]:

$$I(\vec{r}) = \frac{cn\varepsilon_r}{2} |E(\vec{r})|^2 \tag{4-16}$$

So, the local optical intensity $I(\vec{r})$ is proportional to the absolute square of the complex amplitude. In all further context referring to the intensity the factor $\frac{cn\varepsilon_r}{2}$ will be omitted for simplicity, in accordance with:

$$I(\vec{r}) \propto |E(\vec{r})|^2 \tag{4-17}$$

The fact that the relation between the absolute square of the complex amplitude and the intensity is often ignored, is due to their proportional relation. For the calculation of the absolute amount of optical power transmitted exist other methods, that are more suitable than methods based on thin-element approximations. Also, one might differentiate between the intensity as power density $\left[\frac{W}{m^2}\right]$ defined in (4-16) and a relative intensity in arbitrary units ($[a.u.]$) defined in (4-17).

4.1.3 Fresnel-Kirchhoff's Formulation of Diffraction

The description of different scenarios of diffracted waves in space is by all means no trivial task when only using equation (4-13), because it is a differential equation without defined boundary conditions. Figure 4-4 shows a general case, where the problem is to find the wave amplitude at any point P_0 inside the volume V, when a wave disturbance at another point P_1 is induced. Basis of most solutions of diffraction problems is Green's theorem. Let $U(P)$ and $G(P)$ be complex functions of position $P(x,y,z)$, S a closed surface surrounding a Volume V and $\vec{n}$ the normal vector on the surface S. If $U(P)$, $G(P)$ as well as their first and second partial derivatives are continuous on S and single valued than Green's theorem is [32, pp 39, 59]:

$$\iiint_V (U\nabla^2 G - G\nabla^2 U)dV = \iint_S \left(U \cdot \frac{\partial G}{\partial n} - G \cdot \frac{\partial U}{\partial n}\right) dS \tag{4-18}$$

Where $G(P)$ is Green's function, serving as an auxiliary function. Green's function is used to probe the disturbance within the volume V through the boundary along the surface S. According to Huygens's principle, a Green's function in the form of a spherical wave (4-15) is chosen for getting the Kirchhoff's formulation of the diffraction problem [32, pp 40]. So, $G(P_1)$ at an arbitrary point P_1 with the distance to P_0 of r_{01} might be an elementary spherical wave in the form of:

$$\boxed{G(P_1) = \frac{1}{r_{01}} \cdot e^{ikr_{01}}} \tag{4-19}$$

The use of (4-19) creates a singularity problem at $r_{01} = 0$. To solve this problem a spherical section S_ε of S with radius ε is excluded from the integration, so that $S' = S + S_\varepsilon$ is the total surface and $V' = V - V_\varepsilon$ is the new volume. The integration volume and surfaces are illustrated in Figure 4-4. From chapter 4.1.2 we know that (4-15) and therefore (4-19) satisfy the Helmholtz equation (4-13), thus:

$$(\nabla^2 + k^2)G = 0 \tag{4-20}$$

and by definition (of any wave field satisfying (4-13)):

$$(\nabla^2 + k^2)U = 0 \tag{4-21}$$

Inserting (4-20) and (4-21) into the left side of Green's theorem (4-18) yields:

$$0 = -\iiint_{V'} (U\,Gk^2 - GUk^2)\,dV = \iint_{S'} \left(U\frac{\partial G}{\partial n} - G\frac{\partial U}{\partial n}\right) dS \tag{4-22}$$

and because of $S' = S + S_\varepsilon$ and equation (4-22) follows:

$$\iint_S \left(U\frac{\partial G}{\partial n} - G\frac{\partial U}{\partial n}\right) dS = -\iint_{S_\varepsilon} \left(U\frac{\partial G}{\partial n} - G\frac{\partial U}{\partial n}\right) dS \tag{4-23}$$

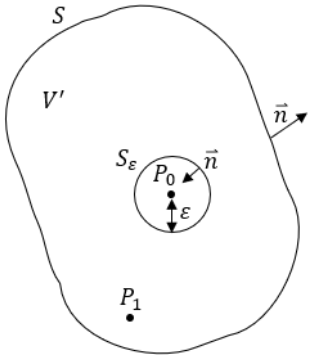


Figure 4-4: Integration Surface S surrounding the integration volume V, around an optical disturbance center P_0. Where $\vec{n}$ are the normal vectors on each surface, S_ε is the surface with radius ε around P_0 to resolve the singularity problem of the Green's function.P_1 is the pointe of the disturbance at an arbitrary point.

Noting that the derivation of G at $P_1 \in V'$ is [32, pp 48, 59]:

$$\frac{\partial G(P_1)}{\partial n} = \frac{1}{r_{01}} e^{ikr_{01}} \left(ik - \frac{1}{r_{01}}\right) \cos(\theta_{01}) \tag{4-24}$$

Where θ_{01} is the angle between $\vec{n}$ and $\overrightarrow{r_{01}}$. Unfortunately, $\frac{\partial G(P_1)}{\partial n}$ is not continuous at $r_{01} = 0$. For solving the singularity problem for a point P_0 outside of S_ε, ε is made arbitrarily small and with (4-24) follows:

$$\lim_{\varepsilon \to 0} \iint_{S_\varepsilon} \left(U\frac{\partial G}{\partial n} - G\frac{\partial U}{\partial n}\right) ds = -4\pi U(P_0) \tag{4-25}$$

Inserting (4-25) into (4-23) results in the Helmholtz-Kirchhoff theorem:

$$\boxed{U(P_0) = \frac{1}{4\pi} \iint_S \left(\frac{\partial U}{\partial n} G - U\frac{\partial G}{\partial n}\right) ds} \tag{4-26}$$

Or:

$$U(P_0) = \frac{1}{4\pi} \iint_S \left(\frac{\partial U}{\partial n}\frac{e^{ikr_{01}}}{r_{01}} - U\frac{e^{ikr_{01}}}{r_{01}}\left(ik - \frac{1}{r_{01}}\right)\cos(\theta_{01})\right) ds \tag{4-27}$$

This result is important because it states that the disturbance at one point within an arbitrary volume can be calculated by knowing the field values at the surface of the volume.

The universal diffraction problem pictured in Figure 4-4 will now be redefined into a more specific case. S is now buildup as the sum of a plane surface S_1 and a spherical section S_2 as shown in Figure 4-5. We assume in this model the radius R of S_2 to be so large that the contribution of $S_2 = S - S_1$ to the Helmholtz-Kirchhoff integral becomes zero. So, for $R \to \infty$ follows $\frac{\partial G}{\partial n} \approx jkG$ and one can argue that with applying the Sommerfeld radiation condition [32, pp 42–44] that, the contribution of S_2 becomes zero, due to a approximately infinite distance of S_2 to the point P_0. The Sommerfeld radiation condition is:

$$\lim_{R\to\infty} R\left(\frac{\partial U}{\partial n} - jkU\right) = 0 \tag{4-28}$$

This states that only disturbances travelling to the outside of S_2 are allowed in this model, so that the contribution of any sources at S_2 remain approximately zero. Equation (4-26) can now be reduced to:

$$U(P_0) = \frac{1}{4\pi}\iint_{S1}\left(\frac{\partial U}{\partial n}G - U\frac{\partial G}{\partial n}\right)ds \tag{4-29}$$

Only the surface S_1 is needed to calculate the field to the right of S_1 in Figure 4-5. The surface S_1 does not have to be flat as pictured. Because in the basic problem of the light field propagation between two planes discussed here, S_1 is assumed to be infinite and perfectly flat.

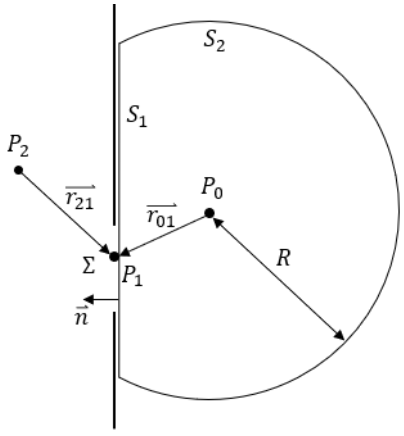


Figure 4-5: Screen with aperture Σ, where the surface S_1 lies directly behind the aperture and has the normal vector $\vec{n}$, the surface S_2 is a spherical section around the observation point P_0 and the radius R. Point P_1 lies on S_1 and inside Σ and P_2 lies left of P_1.

Before proceeding, two assumptions are made to the model of Figure 4-5:

- The field U and its derivative $\frac{\partial U}{\partial n}$ at the aperture surface Σ would be the same as if the screen confining the aperture was not there.

- The field and its derivative at surface S_1 outside of Σ are $U = 0$ and $\frac{\partial U}{\partial n} = 0$.

These two important statements enable the reduction of the integration surface from S_1 to Σ without changing the equation (4-29) and also will not produce significant errors if the wavelength is larger than the aperture. Equation (4-29) might now be written as:

$$U(P_0) = \frac{1}{4\pi} \iint_{\Sigma} \left(\frac{\partial U}{\partial n} G(P_1) - U \frac{\partial G(P_1)}{\partial n} \right) ds \tag{4-30}$$

To simplify things further, only observation points far away, $r_{01} \gg \lambda$, from the aperture are viewed, so $k \gg \frac{1}{r_{01}}$. With this we can approximate equation (4-24) to:

$$\frac{\partial G(P_1)}{\partial n} = \frac{1}{r_{01}} e^{ikr_{01}} \left(ik - \frac{1}{r_{01}} \right) \cos(\theta_{01}) \approx \frac{1}{r_{01}} e^{ikr_{01}} ik \cos(\theta_{01}) \tag{4-31}$$

Finally, if (4-19), (4-31) and $U(P_1) = \frac{e^{ikr_{21}}}{r_{21}}$ for a spherical illumination point are inserted into (4-30),the Fresnel-Kirchhoff (FK) diffraction formulation becomes:

$$\boxed{U(P_0) = \frac{1}{i\lambda} \iint_{\Sigma} \frac{e^{ik(r_{01}+r_{21})}}{2r_{01}r_{21}} [\cos(\theta_1) - \cos(\theta_2)] ds} \tag{4-32}$$

Let us define a special case as an example. The field U is a plane wave approaching the surface S_1 at Σ from the left side in Figure 4-5. P_2 is therefore at infinite distance from the aperture and the incident angle θ_2 is zero. This situation is sketched in Figure 4-6.

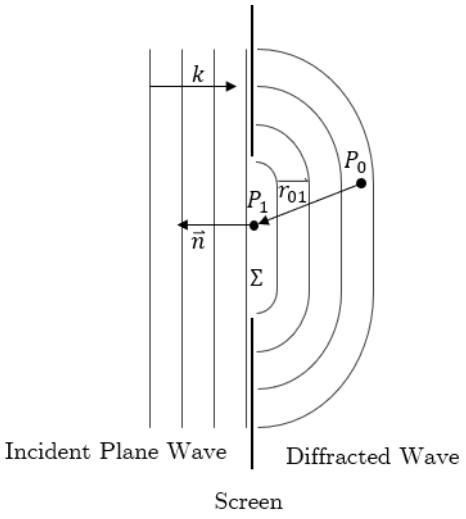


Figure 4-6: Example for the application of the Fresnel-Kirchhoff integral, where a plane wave illuminates the Aperture Σ. The diffracted wave is observed at P_0 with respect to all illuminated points P_1 in Σ.

We can rewrite equation (4-30), (4-31) and use a plane wave (4-14) with inclination angle $\theta_0 = 0$ as U and its derivative $\frac{\partial U}{\partial n} = -ik$ to:

$$U(P_0) = -\frac{i}{2\lambda}\iint_{\Sigma}\frac{e^{ikr_{01}}}{r_{01}}[1+\cos(\theta_1)]ds \tag{4-33}$$

Note that the amplitude A of the plane wave from equation (4-14) is assumed to be one. Although the FK integral produces accurate results when the observation point is many wavelengths away from the aperture, there exist some contradiction in the assumptions made by Kirchhoff. For one, the assumption that the field itself and its derivative cannot be the same as if the screen surrounding the aperture was not there is a contradiction. The presence of a screen will always produce fringing effects [32, pp 44–45], which is accounted for by limiting the position of the observation point to distances much larger than the wavelength. Second, but more important: the statement that the field U and its derivative $\frac{\partial U}{\partial n}$ are both zero also contradicts itself. Only one might be zero at the same time because e.g., if the derivative at the aperture edge would be zero all of the field outside the aperture would be the same as inside the boundaries of the aperture. But if U outside is zero than the field in the aperture would vanish also. [32, pp 46–47, 59]. Removing these inconsistencies will be the topic of the next chapter.

4.1.4 The Rayleigh-Sommerfeld Diffraction Formulation

The Rayleigh-Sommerfeld (RS) integrals resolve the inconsistency of the FK diffraction integral by introducing a new Green's function G, so that (in case of the first RS integral) the change of the field at the aperture by the probing Green's function $\frac{\partial U}{\partial n}G$ becomes zero. This allows us to reduce the boundary conditions from Kirchhoff to be $U \neq \frac{\partial U}{\partial n}$ at the aperture. When using the Green's function [32, pp 47, 60]:

$$\boxed{G_{\pm}(P_1) = \frac{e^{ikr_{01}}}{r_{01}} \pm \frac{e^{ikr'_{01}}}{r'_{01}}} \tag{4-34}$$

Where equation (4-34) represents two cases for subtracting or adding the right term and left term. And the partial derivative of the negative case and $r_{01} = r'_{01}$:

$$\frac{\partial G_{-}(P_1)}{\partial n} = 2\cos(\theta_1)\left(ik - \frac{1}{r_{01}}\right)\frac{e^{ikr_{01}}}{r_{01}} = 2 \cdot \frac{\partial G(P_1)}{\partial n} \tag{4-35}$$

Where it is obvious that (4-35) is just twice the function used in case of Kirchhoff's formulation. One might interpret the case of the negative sign in equation (4-34) as two mirrored probing functions in P_0 and P_0' with a phase shift of 180°. The terms will cancel each other out and equation (4-26) becomes the first Rayleigh-Sommerfeld integral:

$$U_{RS1}(P_0) = -\frac{1}{4\pi}\iint_S U\frac{\partial G_-}{\partial n}ds = -\frac{1}{2\pi}\iint_S U\frac{\partial G}{\partial n}ds \tag{4-36}$$

And analog for the positive case of $G_\pm$ the field U at the aperture becomes zero because $\frac{\partial G_+(P_1)}{\partial n} = 0$ and $G_+(P_1) = 2 \cdot G(P_1)$. So, the second Rayleigh-Sommerfeld integral becomes:

$$U_{RS2}(P_0) = \frac{1}{4\pi}\iint_S G_+\frac{\partial U}{\partial n}ds = \frac{1}{2\pi}\iint_S G\frac{\partial U}{\partial n}ds \tag{4-37}$$

The addition of the mirrored point P_0' is visualized in Figure 4-7, to give a sense for the reduction of Kirchhoff's conditions. The influence of the disturbance from point P_0' cancel each other at the aperture because of a $180°$ phase shift. Either of the RS integrals might be used further. Their application lead to equivalent results. It is common to use the first RS integral for practical reasons, that are that U is often known but not directly its derivative. So, in the following we will work with the first RS integral.

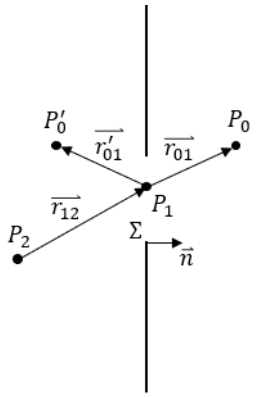


Figure 4-7: Visualization of the diffraction problem with the Green's function $G_\pm$ used by Sommerfeld.

If we now apply the first RS integral (4-36) to the case shown in Figure 4-7 with one single illuminating point at P_2 and the observation point P_0, we insert $G_-(P_1)$ and use the approximation from (4-31) of $k \gg \frac{1}{r_{01}}$, we get:

$$U_{RS1}(P_0) = \frac{1}{i\lambda}\iint_\Sigma U(P_1)\cdot\frac{e^{ikr_{01}}}{r_{01}}\cos(\theta_1)\,ds \tag{4-38}$$

With the diverging source of:

$$U(P_1) = \frac{1}{r_{12}} e^{ikr_{12}} \tag{4-39}$$

We get a comparable form of (4-38) to FK formulation in (4-32):

$$U_{RS1}(P_0) = \frac{1}{i\lambda} \iint_{\Sigma} \frac{e^{ik(r_{01}+r_{12})}}{r_{01} r_{12}} \cos(\theta_1)\, ds \tag{4-40}$$

The problem of $U = 0$ and $\frac{\partial U}{\partial n} = 0$ is solved. But it is worth noting that the RS integrals are only defined for planar screens, otherwise the symmetry of P_0 and P_0' is lost. This is not the case in the FK formulation [32, pp 50–52].

When analyzing the first RS integral further, it is obvious that the integrant might be separated into two parts, like shown here:

$$U_{RS1}(P_0) = \iint_{\Sigma} U(P_1) \cdot \underbrace{\frac{1}{i\lambda} \frac{e^{ikr_{01}}}{r_{01}} \cos(\theta_1)}_{h(P_0,P_1)}\, ds = \iint_{\Sigma} U(P_1) h(P_0, P_1) ds \tag{4-41}$$

When looking at the right part of (4-41) we see a convolution of the field in the aperture with a function describing the function of a system. In our case this function (regardless which function chosen from the above described) is in its characteristic a diverging spherical wave with an amplitude proportional to the radius of the disturbance center. In other words: The field at the observation point is a superposition of spherical waves proportional to the amplitude at the center of each wave and dependent on the distance to the observation point. This describes basically the Fresnel-Huygens principle. It has been already been hinted at in Figure 4-6. Based on equation (4-41) we will now take a look at linear systems and their transfer functions to find an approach consistent with the Rayleigh-Sommerfeld and Fresnel-Huygens description from above.

4.1.5 Linear Systems & Transfer Functions

The linear system theory is commonly used in the field of electronics but can also be applied to optical systems, given some approximations and boundary conditions. The fundamental functionality of a system is: mapping a set of input functions to a set of output functions. Let $u(x, y)$ be an input function of the spatial coordinates $x, y \in \mathbb{R}^2$ and $g(x, y)$ be the output function. u represents the optical field at the aperture and g the field at a given distance from the aperture.

The function of a system transforming u into g is defined as $S\{\,\}$ so that:

$$g(x,y) = S\{u(x,y)\} \tag{4-42}$$

An analogy to system theory approaches, where complex functions are symbolized by blocks to describe information flow in systems, is sketched in Figure 4-8.

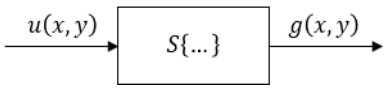


Figure 4-8: Block symbol of system with one input $u(x,y)$ and output $g(x,y)$

The basic properties of a linear system are the linearity between the input and output and the shifting property of the function S [32, pp 18–19]. Linearity is given if the following statement holds true:

$$S\{a \cdot u_1(x,y) + b \cdot u_2(x,y)\} = a \cdot S\{u_1(x,y)\} + b \cdot S\{u_2(x,y)\} \tag{4-43}$$

Where the shifting property is described by:

$$u(x,y) = \iint_{-\infty}^{\infty} u(\xi,\eta)\delta(x-\xi, y-\eta)d\xi d\eta \tag{4-44}$$

The function δ describes the delta function where $\delta(x,y) = \lim_{a\to\infty} a^2 e^{-a^2\pi(x^2+y^2)}$ [32, pp 5]. The first assumption of a systems' linearity allows to find an elementary function mapping u to g, giving the system a response function to an arbitrary input. The shifting property allows any signal to be decomposed into a set of delta functions with u being a weighting factor $u(\xi,\eta)$. When (4-44) is substituted into(4-42) and the linearity theorem from (4-43) is applied, the result takes the form:

$$g(x,y) = \iint_{-\infty}^{\infty} u(\xi,\eta) \cdot S\{\delta(x-\xi, y-\eta)\}d\xi d\eta \tag{4-45}$$

Further, if:

$$h(x,y;\xi,\eta) = S\{\delta(x-\xi, y-\eta)\} \tag{4-46}$$

is the impulse response of a system, we can define (4-45) as a convolution of u with the impulse response h and show that a system under the assumptions of linearity and shift-invariance is only characterized by its impulse response function.

Equation (4-45) therefor becomes:

$$g(x,y) = \iint_{-\infty}^{\infty} u(\xi,\eta) \cdot h(x,y;\xi,\eta) d\xi d\eta \equiv u(\xi,\eta) * h(x-\xi, y-\eta) \tag{4-47}$$

In optics the impulse response function is commonly called point spread function (PSF). The $*$-operator indicates the convolution operation. An interpretation of the convolution might be that every point in g is the superposition of the multiplication of u with h at every position ξ and η. At this point it is worth noting that the superposition theorem of elementary wave by Huygens and Fresnel, shown in the first RS integral (4-41), shows remarkable similarities to (4-45). Besides the convention, that in optics h is called PSF, in the following we use the free-space response function (FSR), because of the analogy to the linear system approach and the actual meaning.

The thorough reader might be interested why the simplification of $h(x,y;\xi,\eta) \equiv h(x-\xi, y-\eta)$ in (4-46) can be done. If we assume a space invariant system, the function h only depends on the distance from excitation point to response point. In an optical sense: We assume that the medium between any source point and the observation point is homogenous. FSR is therefore independent of the spatial position of the observation point or the source of disturbance.

We will now make use of the Fourier transformation to calculate the convolution. If u is a function of the spatial coordinates x and y then its Fourier transformation $\mathcal{F}\{u\} = U$ is a function of the spatial frequencies ν_x and ν_y. Formally written as [32, pp 4–6, 54, pp 1128–1129]:

$$U(\nu_x,\nu_y) = \mathcal{F}\{u(x,y)\} = \iint_{-\infty}^{\infty} u(x,y) \cdot e^{-i2\pi(\nu_x x + \nu_y y)}\, dx\, dy \tag{4-48}$$

And the inverse transformation is described by:

$$u(x,y) = \mathcal{F}^{-1}\{U(\nu_x,\nu_y)\} = \iint_{-\infty}^{\infty} U(\nu_x,\nu_y) \cdot e^{i2\pi(\nu_x x + \nu_y y)}\, d\nu_x\, d\nu_y \tag{4-49}$$

It is widely known that there are many theorems associated with the Fourier transformation and Fourier analysis.

Here are the basic five theorems listed[32, pp 8]:

- *Linearity theorem*

$$\mathcal{F}\{a \cdot u + b \cdot g\} = a \cdot \mathcal{F}\{u\} + b \cdot \mathcal{F}\{g\} \tag{4-50}$$

- *Similarity theorem*

$$\mathcal{F}\{u\} = U \;\rightarrow\; \mathcal{F}\{u(a \cdot x, b \cdot y)\} = \frac{1}{|ab|} U\left(\frac{\nu_x}{a}, \frac{\nu_y}{b}\right) \tag{4-51}$$

- *Shift theorem*

$$\mathcal{F}\{u\} = U \;\rightarrow\; \mathcal{F}\{u(x-a, y-b)\} = U(\nu_x, \nu_y) e^{-i2\pi(\nu_x a + \nu_y b)} \tag{4-52}$$

- *Parseval theorem*

$$\mathcal{F}\{u\} = U \;\rightarrow\; \iint_{-\infty}^{\infty} |u(x,y)|^2 \, dx \, dy = \iint_{-\infty}^{\infty} |U(\nu_x, \nu_y)|^2 \, d\nu_x \, d\nu_y \tag{4-53}$$

- *Convolution theorem*

$$\mathcal{F}\{u\} = U \,; \mathcal{F}\{h\} = H \;\rightarrow\; \mathcal{F}\{u(\xi,\eta) * h(x-\xi, y-\eta)\} = U(\nu_x, \nu_y) H(\nu_x, \nu_y) \tag{4-54}$$

Especially the convolution theorem is of interest. The theorem states that the convolution of two functions u and h is equivalent to a simple multiplication of the Fourier-transformed functions U and H. Thus, an invariant linear system is characterized by its impulse response function h. H is the Fourier transformation of h and is called the transfer function of the system. The output of a system g, corresponding to a specific input, can be calculated by the convolution of the input function u and the impulse response function h or by a multiplication of the input spectrum $U = \mathcal{F}\{u\}$ with the transfer function $H = \mathcal{F}\{h\}$.

4.2 The Angular Spectrum Method

The angular spectrum (AS) method is the logical consequence when applying the linear system theory to the scalar wave description from chapter 4.1. The Fourier transformation of a complex field is called the angular spectrum. The composing nature of the Fourier transformation produces a spectrum of plane waves, travelling in different directions away from the transformation plane. This spectrum of waves in total reflects the field distribution in spatial coordinates exactly if $d\nu_x$ and $d\nu_y$ are sufficiently small, which is not a problem for analytical approaches, but we will see later on that for numerical calculations this must be considered. Nevertheless, in the AS approach each decomposed plane wave is propagated separately by a

phase shift. This phase shift depends on the travelling distance of each wave from the excitation point to the observation point. First, the essential equations used here are summarized. The first RS integral (4-36) can be rewritten with $k = \frac{2\pi}{\lambda}$ and the derivation of Green's function from (4-24) as follows:

$$u_{RS1}(P_0) = -\frac{1}{2\pi}\iint_S u \cdot \frac{1}{r} e^{ikr}\left(ik - \frac{1}{r}\right)\cos(\theta)\, ds$$
$$\Rightarrow u(x,y;z) = \iint_S u(\xi,\eta;0) \cdot \frac{1}{r} e^{ikr}\left(\frac{1}{i\lambda} + \frac{1}{2\pi r}\right)\frac{z}{r}\, ds \qquad (4\text{-}55)$$

With $r = \sqrt{(x-\xi)^2 + (y-\eta)^2 + z^2}$. So that the free-space impulse response function $h_{RS}(x,y;z)$ according to the first RS integral becomes:

$$h_{RS}(x,y;z) = \frac{1}{r} e^{ikr}\left(\frac{1}{i\lambda} + \frac{1}{2\pi r}\right)\frac{z}{r} \qquad (4\text{-}56)$$

By calculating the Fourier transform of (4-56) the free-space transfer function or free-space propagation function (FSP) becomes [32, pp 60–61, 54, pp 111, 61, 62]:

$$H_{RS}(\nu_x,\nu_y;z) = \mathcal{F}\{h_{RS}(x,y;z)\} = e^{i2\pi z\sqrt{\frac{1}{\lambda^2}-\nu_x^2-\nu_y^2}} \qquad (4\text{-}57)$$

Where caution is advised, because for spectral components where $\nu_x^2 + \nu_y^2 > \frac{1}{\lambda^2}$, the square root in the exponent of equation (4-57) becomes imaginary and the overall transfer function becomes real valued. The transfer function then represents an attenuation factor $e^{-2\pi z\sqrt{\nu_x^2+\nu_y^2-\frac{1}{\lambda^2}}}$ [32, pp 57–59, 54, pp 111–112]. The physical meaning of this attenuation is, that spatial frequencies higher than $\frac{1}{\lambda}$ represent evanescent fields which are dropping rapidly in amplitude close to the aperture. The contributions to the field distribution a few wavelengths away from the aperture can be ignored, as spatial frequencies over $\frac{1}{\lambda}$ are viewed as a cutoff frequency of the free-space system. A circle function is defined to limit the spatial frequency bandwidth to the radius r as:

$$circ(r) = \begin{cases} 1, r \le 1 \\ 0, r > 1 \end{cases} \qquad (4\text{-}58)$$

With the limitation of (4-58) the transfer function H_{RS} (4-57) is defined as:

$$\boxed{H_{RS}(\nu_x,\nu_y;z) = e^{i2\pi z\sqrt{\frac{1}{\lambda^2}-\nu_x^2-\nu_y^2}} \cdot circ\left(\lambda^{-2}\sqrt{\nu_x^2+\nu_y^2}\right)} \qquad (4\text{-}59)$$

It has been shown by Sherman [63] that the AS method using the RS kernel H_{RS} yields equivalent result as a RS direct integration method of the integral itself. To summarize, the AS method calculates the complex amplitude field distribution of a wave $g(x,y;z)$ as a superposition of planar waves at a plane at z, parallel to the aperture plane at $z = 0$ with the field distribution $u(\xi,\eta;0)$ by the following relation:

$$\boxed{g(x,y;z) = \mathcal{F}^{-1}\left\{H_{RS}(\nu_x,\nu_y;z)\cdot\mathcal{F}\{u(\xi,\eta;0)\}\right\}} \tag{4-60}$$

4.2.1 The Discrete Fourier Transformation of Sampled Fields

The analytical method of the Fourier transformation of equation (4-48) and (4-49) gives a understanding of how the convolution is connected to the spectral multiplication, but is not practical for real calculations. Real measured data is always sampled and by using matrix operations all data and operators are sampled as well. So, the conclusion is that a discrete Fourier transformation (DFT) is needed to represent the spectra of the field distributions. First, the properties of our sampled fields have to be defined. For this, the continuous field is represented by a set of delta functions δ, shifted by a constant offset Δx, Δy or $\Delta\xi$, $\Delta\eta$ in each direction. A one-dimensional visualization is shown in Figure 4-9, where a continuous signal $u(x)$ is sampled by discrete values with a spacing of Δx.

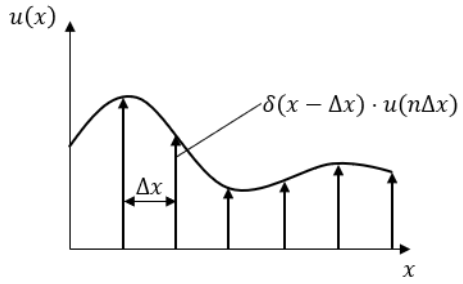


Figure 4-9: Discretization of a one-dimensional continuous signal u with shifted δ-functions by Δx and weighted with the local amplitude at $n\Delta x$.

In the two-dimensional case, the function $u(x,y)$ is sampled by a grid of delta functions shifted by Δx and Δy. A sampled function or vector field $\boldsymbol{u}$ can be described by taking each value of the spatially separated sample point of the delta function grid:

$$\boldsymbol{u} = \underbrace{\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\delta\left(\frac{x}{\Delta x}-n\right)\delta\left(\frac{y}{\Delta y}-m\right)}_{Sampling\ Grid}\cdot u(x,y) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}u(n\Delta x, m\Delta y) \tag{4-61}$$

In equation (4-61) the variables n and m are integers. It is further assumed that the function u is non-zero only in a finite area, so the sampled field $\vec{u}$ can be defined over a finite area with

the size of $M\Delta x \cdot N\Delta y$. The arrow notation of $\vec{u}$ indicates a discretized vector. Everywhere outside of this area the function is assumed to be zero.

$$\vec{u} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} u(n\Delta \mathrm{x}, m\Delta y) \tag{4-62}$$

Here, the absolute size of the sampled area is the multiplication of the total number of points in each direction N and M with the spacing between each sampling point Δx and Δy. The approximation from (4-62) contradicts the wave nature of the fields of interest but is also a consequence of finite computing power in reality. This, in turn, has consequences when using the DFT. The two-dimensional DFT is defined as [64]:

$$\vec{U} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \vec{u} \cdot e^{-i2\pi \frac{mk}{M}} e^{-i2\pi \frac{nl}{N}} = DFT(\vec{u}) \tag{4-63}$$

The inverse two-dimensional DFT is defined as:

$$\vec{u} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \vec{U} \cdot e^{i2\pi \frac{mk}{M}} e^{i2\pi \frac{nl}{N}} = IDFT(\vec{U}) \tag{4-64}$$

k and l are the frequency bins. The assumption made in (4-62), of a finite signal and the periodic nature of the Fourier series approximation by the DFT, produce aliases, a.k.a. replicas in the frequency domain and when back transformed with the IDFT also aliases in the spatial domain. The aliases in the frequency domain are exactly $1/\Delta x$ apart. To compute the DFT, typically the Cooley-Tukey Fast Fourier transform algorithm (FFT) is used [64]. It exploits symmetries of the DFT to minimize the computational effort needed from $\mathcal{O}_{DFT} = (N \cdot M)^2$ to $\mathcal{O}_{FFT} = (N \cdot M) \cdot \log(N \cdot M)$. The FFT algorithm produces exactly the same output as the DFT itself and will be used in place of the DFT in all further mentions, so the two abbreviations are completely interchangeable. The FFT produces a two-sided spectrum with a shifted zero frequency. As shown in the not zero-shifted spectrum in Figure 4-10 a), the lower half of the alias is shifted by $-\frac{1}{\Delta x}$ to negative spatial frequencies in the one-dimensional case to yield the zero-shifted spectrum shown in Figure 4-10 b). In two dimensions the zero-shift can be realized by dividing the unshifted field in to four quadrants and switch the quadrant 1 with 4 and 2 with 3. The 2D-zero-shift is illustrated in Figure 4-10 c).

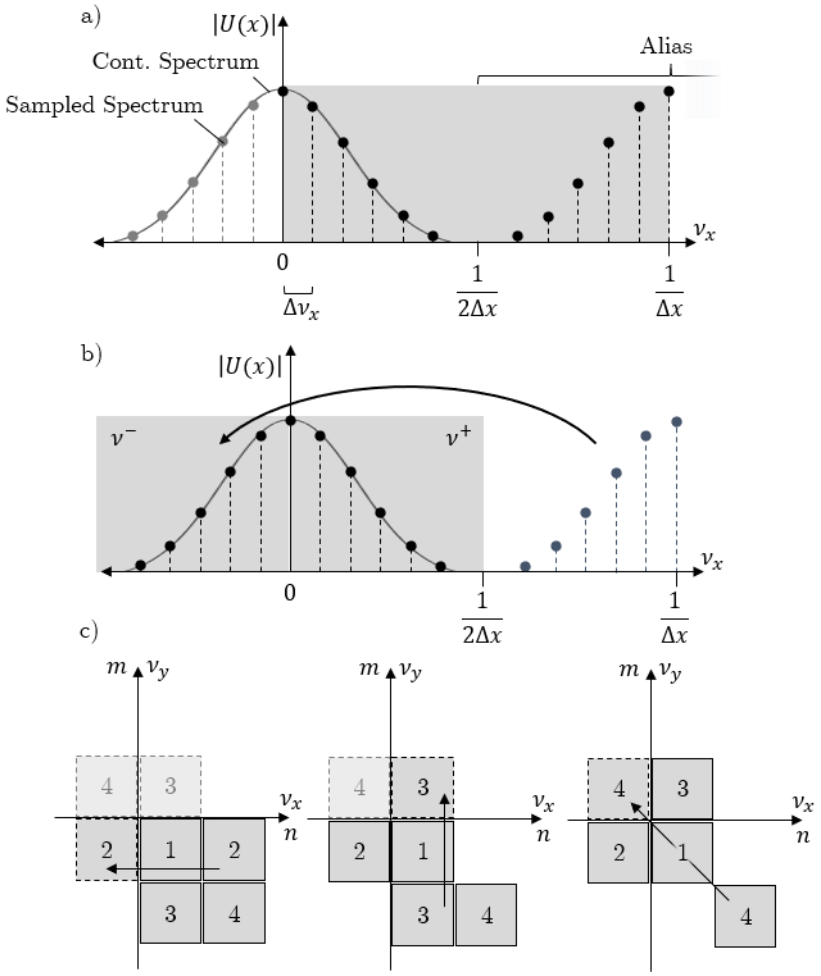


Figure 4-10: a) Amplitude output of a one-dimensional DFT/FFT operation with the alias centered at the sampling frequency and the Nyquist Frequency $1/2\Delta_x$. In b) the corrected spectrum with the lower frequency half of the alias as negative frequency range ν^-. The continuous spectrum of a cont. Fourier transform is indicated as a solid line and the sampled spectrum as dots representing the power in each frequency bin of width ν_x and c) shows the zero-order shifting for a two-dimensional FFT spectra in three steps.

The FFT produces a spectrum with discrete frequency bins of width $\Delta\nu_x = \frac{1}{L_x}$ and the signal length is $L_x = N\Delta x$. The frequency range in ν_x-direction ranges from $-\frac{1}{2\Delta x}$ to $\frac{1}{2\Delta x}$, which stems from the Nyquist-Shannon sampling theorem [65].

4.2.2 Zero-Padding for Discrete Linear Convolutions

When propagating a field by convolution with a free-space kernel according to (4-60), the linear convolution, in case of the continuous transformed fields, becomes a circular convolution because the FFT assumes periodicity in the spectral and spatial domain. The aperture function itself is by definition of the RS integral, inherited by the Kirchhoff integral from (4-30), zero outside of the aperture opening Σ. The periodicity contradicts this assumption. By this violation artefacts are introduced in the propagated field called wraparound [66]. To minimize this error,

the calculation window can be increased, so the signal of the aperture itself becomes more aperiodic. In terms of discrete vectors, this process is accomplished by adding zeros at the computation window edges. The addition of zeros i.e., zero-padding, increases the spectral resolution $\Delta\nu = 1/L$, where L is signal length, and when applying the inverse FFT, aliases in the spatial domain (replicas) have an increased spacing $\Delta x_{\mathrm{Rep}} = 1/\Delta\nu$. The wraparound effect and effect of adding a padding around the aperture are shown in Figure 4-11. The errors of the circular convolution will never vanish completely, but the zero-padding method is a sufficient way to linearize the convolution operation, at expense of computational power.

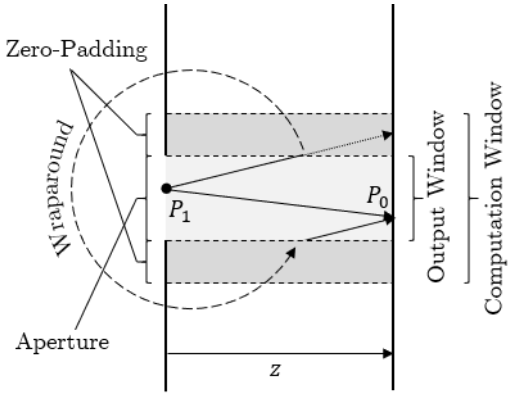


Figure 4-11: Wraparound contributions at P_0 due to no zero-padding. If zero-padding is added to the computation window the contribution indicated by the dotted arrows falls onto the screen and not onto P_0.

The angular spectrum of the input vector $\vec{u}$ with the addition of the padding might be written as follows:

$$U_P = FFT\{u_P\} = FFT\left\{\begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \vec{u} & \vdots \\ 0 & \cdots & 0 \end{bmatrix}\right\} \tag{4-65}$$

Here, the convention of the discretized vector notation $\vec{x}$ is dropped for simplicity and just x is used instead, because all further calculations assume a finite sampled data set. By increasing the computational window by a zero-padding, the total number of sampling points increases and therefore the spectral resolution of the Fourier transformed field increases, similar as stated above:

$$L_{x,P} = \Delta x(N_x + 2N_P) = \frac{1}{\Delta\nu_P} \tag{4-66}$$

The amount of padding necessary can be derived by using a ray model of the calculation window and adjacent replicas. It has been proven, that the local spatial frequencies of the free-space transfer function can be associated with the ray direction of each plane wave component

[32, pp 15-18,442-443]. The local spatial frequencies refer to the varying phase amplitude of the FSP image, rather than the actual transformed spatial frequency. To emphasize this, Figure 4-12 shows the phase amplitude of a transfer function. To the outer edges the phase amplitude oscillates faster than in the center area. It is noteworthy that the correlation of local spatial frequencies and the components' travelling direction is only valid for components where the phase amplitude varies linearly with the spatial frequency, which is not true for outer spectral components if the propagation distance is large.

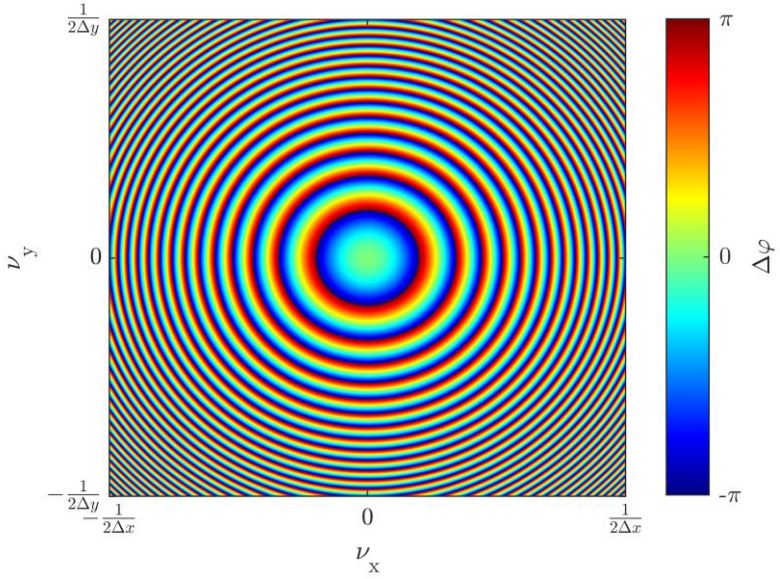


Figure 4-12: Phase shift of a free-space transfer function $H(\nu_x, \nu_y; z)$ at $z = 100\,\lambda$ for a wavelength of $\lambda = 1$

The cosines of the ray directions of each frequency component are the local spatial frequencies of the phase amplitude distribution and can be found according to Goodman [32, pp 16] by:

$$\theta_x = \cos^{-1}(\lambda f_{\nu_x}) = \cos^{-1}\left(\lambda \cdot \frac{1}{2\pi} \cdot \frac{\partial}{\partial \nu_x}\varphi(\nu_x, \nu_y; z)\right) \tag{4-67}$$

and:

$$\theta_y = \cos^{-1}(\lambda f_{\nu y}) = \cos^{-1}\left(\lambda \cdot \frac{1}{2\pi} \cdot \frac{\partial}{\partial \nu_y}\varphi(\nu_x, \nu_y; z)\right) \tag{4-68}$$

where $\varphi(\nu_x, \nu_y; z)$ is the phase amplitude of $H(\nu_x, \nu_y; z)$, which is just the argument of H. The spectral components of the replicas outside of the aperture field travel at the same angles towards the observation plane. If no sufficient padding for the aperture window is chosen, it is possible for spectral components of the replicas to fall into the observation window. The condition for spectral components to be distinguishable from replicas is shown in Figure 4-13. Hereby the aperture window size L_x is the same as the observation window size therefore, one

can conclude that, at a specific observation distance z, the highest local spatial frequency needed travels at the angle $\theta_{Max} = arctan\left(\frac{L_x}{z}\right)$ which is the same as the maximum replica local spatial frequency angle $\theta_R = \max(\theta_x)$ from equation (4-67). With $\theta_{Max} \equiv \theta_R$, it becomes obvious that a vector connecting the furthest apart point of the aperture window with the point on in the observation window does not overlap with the vector at the same angle of the replica if:

$$L_{x,P} \geq 2 \cdot L_x \tag{4-69}$$

The case where the condition of (4-69) is met is shown in Figure 4-13 b). When evaluating a propagating field, the padded regions of the input field have to be ignored in the output field, as they contain wraparound effects. The inequality (4-69)gives a minimum requirement for the AS method. But for propagation distances much larger than the size of the aperture window an increasing error is introduced [61, 67] because the FSP function suffers from aliasing errors itself. A method to handle these errors is topic of the next chapter.

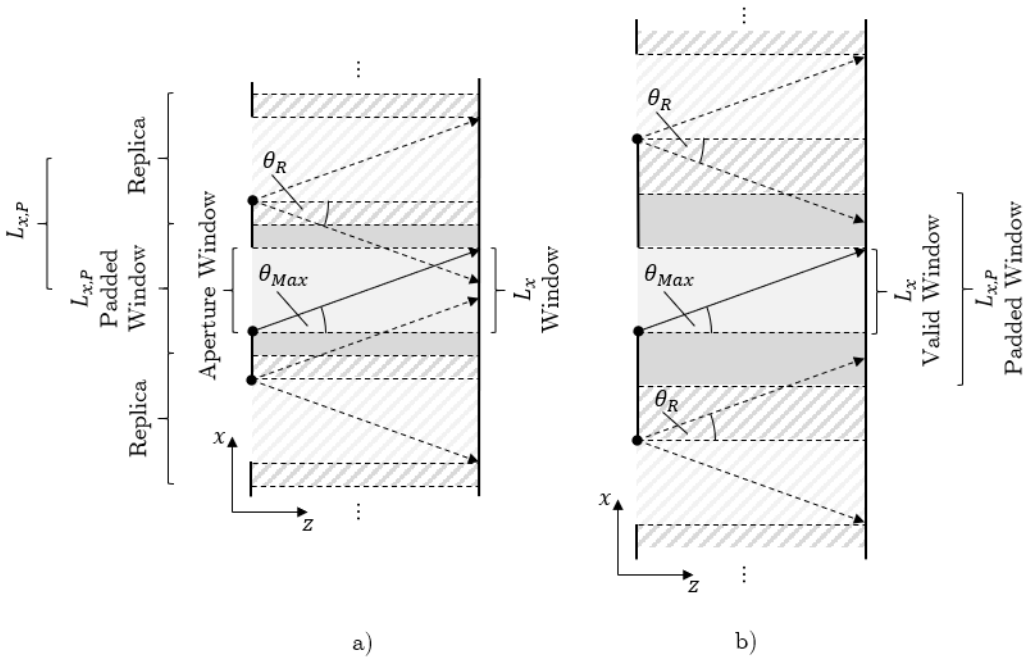


Figure 4-13: Geometrical description of an aperture field propagated to a screen with a) insufficient padding so that power of the replicas influence the output field in the computation window and b) a sufficient padding so that no spectral components influence the field at the observation plane in a specific observation window.

4.2.3 Band-limited Angular Spectrum Method (BLAS)

As stated in the previous chapter 4.2.2: The zero padding is a way to linearize the circular convolution and avoids the spilling of optical power from replicas into the computational window. However, a FSP function calculated in the frequency domain according to (4-59) is not band-limited and therefore might be subject to aliasing errors, too. Especially with increasing observation plane distance, the higher frequency components in the spectrum of the FSP function increase. An example of the influence of the padding size is shown in Figure 4-14, where a) is an oversampled FSP function so that almost no aliasing effect is visible and b) is an undersampled FSP function so that higher local spatial frequencies of higher spectral frequencies appear lower than in a). In other words: In equations (4-67) and (4-68) was established that the cosine of the physical angles at which the discrete spectral components travel are the local spatial frequencies f_{ν_x} and f_{ν_y} multiplied by λ. These local spatial frequencies appear to become slower with higher spectral frequencies in Figure 4-14 b) than in a). The result is that higher spectral frequency components appear to travel at smaller angles and introduce simulation errors although the criteria from (4-69) is met.

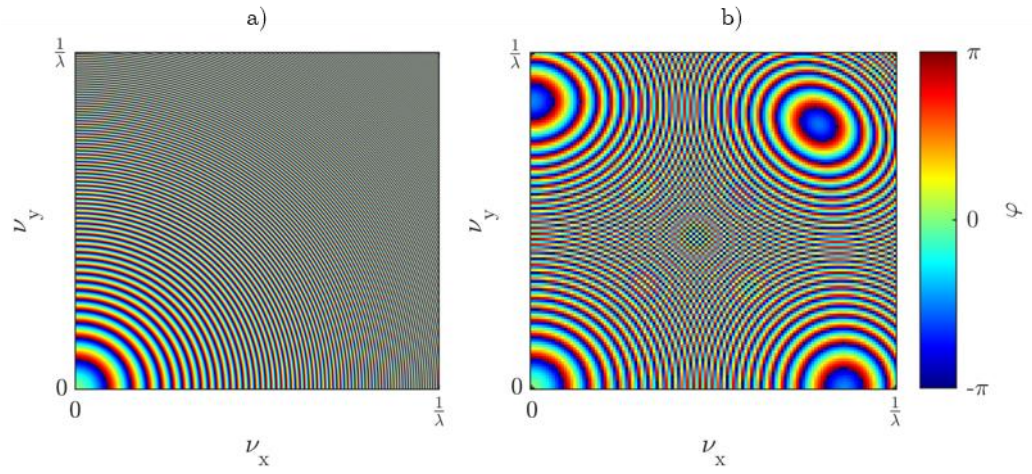


Figure 4-14: Phase of FSP functions at $z = 500\,\lambda$ and $\lambda = 1$ with a) oversampled at $\Delta\nu_{x,y} = 50 \cdot 10^{-3}\,\frac{1}{\lambda}$ ($2000^2\,px$) and b) undersampled at $\Delta\nu_{x,y} = 8.3 \cdot 10^{-3}\,\frac{1}{\lambda}$ ($120^2\,px$), where the zero-frequency is in the bottom-left corner

The maximum local spatial frequency in H increases with the propagation distance. From the equations (4-67) and (4-68) one can isolate the term for the local spatial frequencies in ν_x and ν_y direction:

$$f_{\nu x} = \frac{1}{2\pi} \cdot \frac{\partial}{\partial \nu_x} \varphi(\nu_x, \nu_y; z) \tag{4-70}$$

Where the focus lies on the treatment of the x-direction for simplicity. The calculation for the y-direction is the same with respect to the spectral frequency ν_y. With the derivative of the phase amplitude $\frac{\partial}{\partial \nu_x}\varphi(\nu_x, \nu_y; z) = \frac{\partial}{\partial \nu_x}\left(2\pi z\sqrt{\frac{1}{\lambda^2} - \nu_x^2 - \nu_y^2}\right)$ with respect to ν_x equation (4-70) becomes:

$$f_{\nu x} = \frac{-z\nu_x}{\sqrt{\frac{1}{\lambda^2} - \nu_x^2 - \nu_y^2}} \tag{4-71}$$

Following the Nyquist theorem of signal sampling [65] the bandwidth required to sample the FSP function without aliasing errors is:

$$\frac{1}{\Delta\nu_x} \geq 2 \cdot \max(|f_{\nu x}|) \tag{4-72}$$

Where the maximum values of $f_{\nu x}$ are assumed to be at the highest spatial frequencies, based on a qualitative evaluation of Figure 4-14. The highest positive spatial frequency of the FPS function is at $\nu_x = \frac{1}{2\Delta x}$. So, in respect to the x-direction and based on the relation (4-72) the sampling relation for calculating the FSP function H without aliasing of the phase amplitude can be defined as:

$$\boxed{L_{x,P} \equiv \frac{1}{\Delta\nu_x} \geq z\left(\Delta x\sqrt{\lambda^{-2} - (2\Delta x)^{-2} - (2\Delta y)^{-2}}\right)^{-1}} \tag{4-73}$$

Where $L_{x,P} = L_x + 2L_P$ is the computational window size with padding, z is the distance from the aperture plane to the observation plane, λ is the wavelength, Δx and Δy are the spatial sampling distances of the aperture and observation plane. Similar, the sampling condition for frequency components in y-direction can be derived:

$$\boxed{L_{y,P} \equiv \frac{1}{\Delta\nu_y} \geq z\left(\Delta y\sqrt{\lambda^{-2} - (2\Delta x)^{-2} - (2\Delta y)^{-2}}\right)^{-1}} \tag{4-74}$$

The obvious approach to avoid aliasing, according to (4-73) and (4-74), when a aperture size L_x, L_y and wavelength λ are given, is to further increase padding size $L_{x,P}$ and $L_{y,P}$ than demanded by (4-69). This might work up to a certain degree, but this method is limited by the increasing computational power needed by the increasing of the window size. Another possibility is to reduce the spatial resolution by increasing the sampling spacing Δx and Δy, which is viable also. By doing that, the total sampling number $N_{tot} = \frac{L_{x,P}}{\Delta x}$ and $M_{tot} = \frac{L_{y,P}}{\Delta y}$ of the input additionally decreases depending on the window size L_x and L_y, which saves computational power. As enticing as the reduction of spatial resolution might be, it has a lower

limit which is set by the wanted output resolution or the needed sampling of the aperture window to reflect the aperture function correctly. The dependence of needed computation window size L_x and spatial resolution of the aperture and observation window Δx is shown in Figure 4-15 for different observation distances. The window size L_x increases linear with the distance, but the computational points for a two-dimensional calculation therefore increase with the square of the propagation distance. Hereby is the padding condition from (4-69) as well as the actual aperture window size without padding are ignored. The point is that even with a low spatial resolution $\frac{1}{\Delta x}$ in the aperture window, at higher propagation distances z a larger number for sampling points $N_{tot} \times M_{tot} = \left(\frac{L_x}{\Delta x}\right)^2$ is needed and when higher accuracy of $\Delta x < \lambda$ the aperture is needed, then the necessary window length L_x increases dramatically.

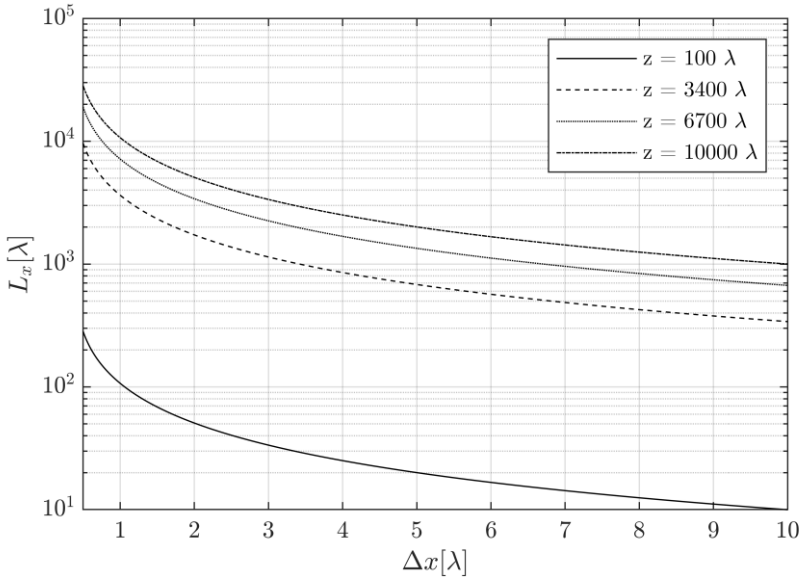


Figure 4-15: Minimal FSP function sampling requirement (4-73) plotted for different observation plane distances z, arbitrary small aperture size and in units of wavelengths.

However, the Band-limited Angular Spectrum (BLAS) method is based on a slightly different approach. For this, equation (4-72) is rearranged to give a maximal spatial frequency $\nu_{x,max}$ so no aliasing of $\arg(H)$ occurs, with $\nu_y = 0$:

$$\frac{1}{\Delta \nu_x} \geq 2 \frac{z \cdot \nu_{x,max}}{\sqrt{\frac{1}{\lambda^2} - \nu_{x,max}^2}} \tag{4-75}$$

Solving (4-75) for $\nu_{x,max}$ yields:

$$\boxed{\nu_{x,max} \leq \frac{1}{\lambda \sqrt{4 \Delta \nu_x^2 z^2 + 1}}} \tag{4-76}$$

Applying the same procedure to the ν_y-direction also yields:

$$\nu_{y,\max} \leq \frac{1}{\lambda\sqrt{4\Delta\nu_y^2 z^2 + 1}} \tag{4-77}$$

Note that both equations give a spatial frequency bandwidth if the other spatial frequency is zero i.e., condition (4-76) for $\nu_{x,max}$ is only valid if $\nu_y = 0$ and vice versa. But according to [61] both conditions are simultaneously met if:

$$\frac{\nu_x^2}{\nu_{x,max}^2} + \frac{\nu_y^2}{\lambda^{-2}} \leq 1 \tag{4-78}$$

and:

$$\frac{\nu_x^2}{\lambda^{-2}} + \frac{\nu_y^2}{\nu_{y,max}^2} \leq 1 \tag{4-79}$$

Both conditions, (4-78) and (4-79), are ellipses in the frequency plane with the minor axis length of $2\nu_{x,max}$ and $2\nu_{y,max}$ respectively. The major axis has a total length of $\frac{2}{\lambda}$. Those ellipses are shown in Figure 4-16 a) and b). The blue ellipse indicates the area in which the local frequency f_{ν_y} components in ν_y direction are sampled correctly and the red ellipse the area for correctly sampled local frequency f_{ν_x} components in ν_x. The overlapping area is, were both conditions (4-78) and (4-79) are fulfilled. If the eccentricity is large enough i.e., $\frac{1}{\lambda} \gg \nu_{x,max}$ and $\frac{1}{\lambda} \gg \nu_{y,max}$ the common region might be approximated by a rectangular area, shown as a green rectangle in Figure 4-16. In Figure 4-16 a) the observation plane distance z is smaller than in Figure 4-16 b).

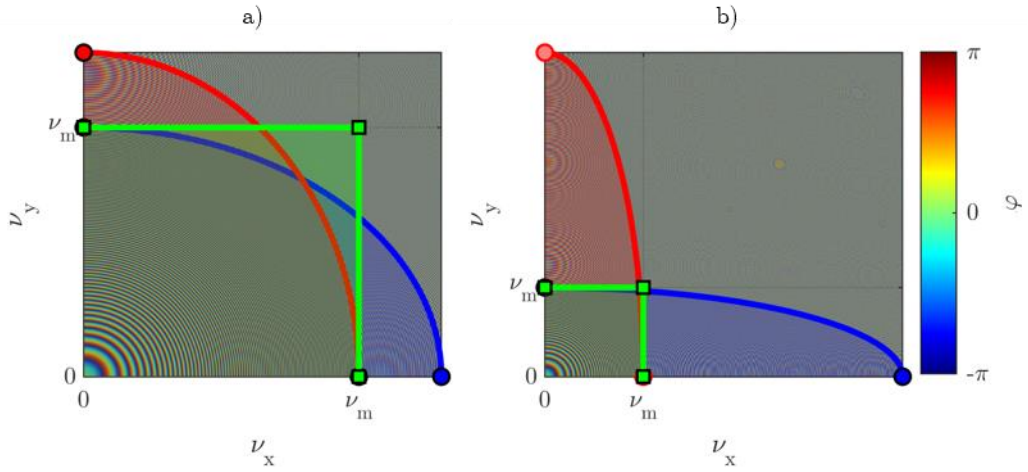


Figure 4-16: Phase amplitude of two FSP functions at a) $1200\,\lambda$ an b) at $3600\,\lambda$. The red ellipse indicates the valid region of ν_x calculated by (4-78). The blue ellipse indicates the valid region for ν_y calculated by (4-79). ν_m is the maximum spatial frequency $\nu_{x,max} = \nu_{y,max}$ for the case of symmetric sampling of the calculation window.

When the distance z increases, the sampling limit decreases, and the eccentricity increases. The rectangular approximation is less accurate for smaller distances, yet the error is less dramatic due to less spectral components getting undersampled.

To exclude undersampled frequency components outside of the green rectangle in Figure 4-16, a window function can be applied to the FSP function H before multiplication with the aperture spectrum. This windowing introduces a bandwidth limitation, hence the name Band-limited Angular Spectrum method. The BLAS method allows more accurate calculation of far-field propagation [61, 67]. The new FSP function H_{BL} with applied window function W_{max} can be described by:

$$\boxed{H_{BL}(\nu_x, \nu_y; z) = H(\nu_x, \nu_y; z) \circ W_{max}(\nu_x, \nu_y; z)} \quad (4\text{-}80)$$

Where the $\circ$-operator is the element-wise multiplication, a.k.a. Hadamard (or Schur) product [68, pp 477–483]. The window function W_{max} in the BLAS approach is:

$$W_{max}(\nu_x, \nu_y; z) = \begin{cases} 1\,, if\ |\nu_x| \le \nu_{x,max}\ \wedge\ |\nu_y| \le \nu_{y,max} \\ 0\,, if\ |\nu_x| > \nu_{x,max}\ \wedge\ |\nu_y| > \nu_{y,max} \end{cases} \quad (4\text{-}81)$$

or:

$$W_{max}(\nu_x, \nu_y; z) = rect\left(\frac{\nu_x}{2\nu_{x,max}}\right) rect\left(\frac{\nu_y}{2\nu_{y,max}}\right) \quad (4\text{-}82)$$

Note that, although in Figure 4-16 only one quadrant of the complete frequency range is shown, the window function of (4-83) must limit the phase amplitude in the negative range for each spatial frequency also, hence the factor of 2. Where the rectangular function is:

$$rect(\chi) = \begin{cases} 0, & |\chi| > \frac{1}{2} \\ \frac{1}{2}, & |\chi| = \frac{1}{2} \\ 1, & |\chi| < \frac{1}{2} \end{cases} \quad (4\text{-}83)$$

In the basic AS method, the undersampled local frequency components cause increasing numerical errors with increasing propagation distance, which is avoided by the truncation of H. A needed bandwidth for accurate calculation exists also. As the limitation of a minimum padding area, the needed bandwidth $\nu_{x,need}$ and $\nu_{y,need}$ can be derived from the geometrical model from Figure 4-13 [61, 67]. The highest angle from the furthest points away from each other in the calculation window of aperture plane and observation plane give the highest local spatial frequencies. In other words, the corner points still falling inside of the green window

function in Figure 4-16 must reach the opposite corner in the observation plane. The points that are responsible for those frequency components are shown in Figure 4-17 and are the beginning and ending points of maximum radius r. Because of the relation of travelling angle and local spatial frequency, the needed spatial frequencies can be derived from window sizes, according to [61, 67], from the geometrical model as follows:

$$\nu_{x,\text{need}} \leq \frac{1}{\lambda\sqrt{\left(\frac{z}{L_x}\right)^2 + 1}} \tag{4-84}$$

For $L_{x,p} = 2L_x$ and identical sizes of the aperture window and observation window. When the window sizes are equal, the needed frequency $\nu_{x,need}$ is the same as the band limit $\nu_{x,max}$ from equation (4-76).

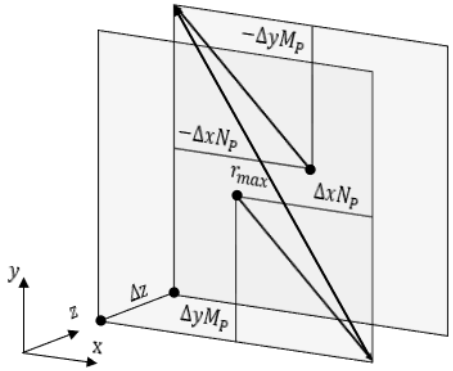


Figure 4-17: Aperture & observation plane with two points with the greatest distance i.e., with the maximal needed local frequency f_{ν_x} and f_{ν_y}

The following list summarizes the sampling conditions for the BLAS method:

- The highest allowable spatial frequency for the FSP function and aperture window is:

$$\boxed{\nu_{\max} \leq \frac{1}{\lambda\sqrt{4\Delta\nu^2 z^2 + 1}}}$$

  All frequency components above ν_{max} are to be clipped by a window function W_{max}.

- The sampling intervals in the spatial domain Δx and Δy are to be chosen so that:

$$\boxed{\Delta s \leq \frac{1}{2\nu_{max}}}$$

  Where Δs is the spatial sampling interval in x- and y-direction. The upper limit for the spatial sampling is given by the aperture function. Δs needs to sample the aperture function within the sampling condition of Nyquist-Shannon [65].

- The lower limit of the padding size of the aperture window is:

$$L_P \geq 2 \cdot L$$

- For large propagation distances z the window function of ν_{max} becomes small and the spectral resolution vanishes as $\nu_{max} \to \Delta\nu$. The size of the padded aperture function is the reciprocal value of the spectral resolution $\Delta\nu = \frac{1}{L_P}$, therefore:

$$L_P = \frac{1}{\Delta\nu} \gg \frac{1}{\nu_{max}}$$

4.3 Complex Amplitude Wavefront Modulation

The propagation of a wavefront described in 4.2 is only one part of the solution of modelling a diffractive layer. Before propagation, the wavefront is locally modulated to reflect the neurons' impact on each incoming input. The modulation is modeled by a thin optical element which acts on the phase and/or amplitude of the input. The physical object itself may be a diffractive optical element, holographic surface, or an obscuration mask. In Figure 4-18 are possible types of modulations shown. A phase-only modulation is sketched in Figure 4-18 a) and b), where a) shows a surface with varying thickness d and a constant refractive n_1 of a thin element. In turn, b) shows a localized change in the refractive index n and a constant thickness d. Both methods are treated equally in the diffraction simulation approach here. The properties of varying scattering due to varying thickness, reflections inside the medium and at vertical "pillars" are omitted by assuming an ideal element with a thickness d of approximately zero to ignore those parasitic reflection and scattering effects:

$$d \to 0 \tag{4-85}$$

However, the phase modulation of either model a) and/or b) is given by:

$$\Delta\varphi(x,y) = \frac{2\pi}{\lambda} \cdot \Delta n(x,y) \cdot \Delta d(x,y) \tag{4-86}$$

In agreement with the scalar wave model of 4.1, polarization effects are omitted as well. Figure 4-18 a) shows how a local change in thickness Δd of a thin plate with constant refractive index n_1 acts as a modulator for wavefront. In turn, Figure 4-18 b) shows a modulation only done by varying the refractive index Δn of a thin plate. Both approaches can be treated equally according to equation (4-86).

Formally, the complex transmission includes phase and amplitude modulation and might be written with the approximation mentioned, as follows:

$$t(x,y) = T(x,y) \cdot e^{-i\Delta\varphi} \tag{4-87}$$

With (4-87), the wave immediately after the modulating element $u(x,y;0^+)$ is:

$$\boxed{u(x,y;0^+) = u(x,y;0^-) \cdot t(x,y)} \tag{4-88}$$

Where $u(x,y;0^-)$ is the wave right before the modulating element. The function $u(x,y;0^+)$ is effectively the aperture function for the BLAS algorithm, describing the complex wave in the aperture itself. Additionally, the complex transmission function t can be calculated to represent specific optical elements like thin lenses, wedges, or prisms. Those elements are defined for one wavelength.

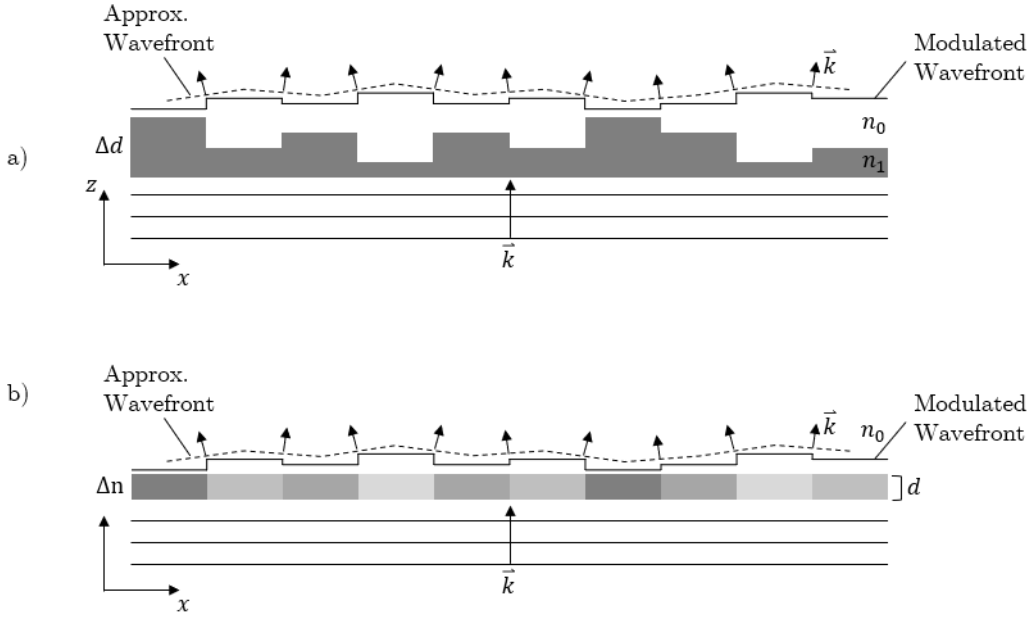


Figure 4-18: Types of wavefront modulations. a) Phase modulation $\Delta\varphi$ by local variations Δd of the travel distance inside a medium of constant refractive index n_1. b) Phase modulation $\Delta\varphi$ by local variations of the refractive index Δn of a medium with constant thickness.

An incoming wavefront $u(x,y;0^-)$ can be modified by a complex transmission function $t(x,y)$, so that, when propagated through space, the wavefront $u(x,y;0^+)$ has the characteristic as if the wave would have travelled through an ideal optical element. The aperture function i.e., complex transmission function of a plano-convex lens might be described by its local optical path difference (OPD). In case of a lens, as shown in Figure 4-19, the OPD depends on the thickness of the lens at the incident point and the refractive index of the material.

The focal length of a lens at a specific radial distance from the optical axis can be calculated by [54, pp 54–55]:

$$f' = \frac{R}{n_1 - n_0} = \frac{R}{\Delta n} \tag{4-89}$$

and, with geometrical relations, the local thickness of the sphere cap becomes:

$$d(r) = d_0 - R + \sqrt{R^2 - r^2} \tag{4-90}$$

Where d_0 is the apex thickness of the lens, $R = f'\Delta n$ and $r = \sqrt{x^2 + y^2}$ is the radial distance from the center. Now, the optical path length (OPL) is the actual length that light travels throught the lens $d(r)$ with respect to the material density n_1. The OPL for a medium with constant refractive index is defined as refractive index multiplied with the with geometrical path length s:

$$OPL = s \cdot n \tag{4-91}$$

And the OPD of each point of the lens is therefore[25]:

$$OPD(x, y) = (n_1 - n_0) \cdot \big(d_0 - d(x, y)\big) = \Delta n \Delta d \tag{4-92}$$

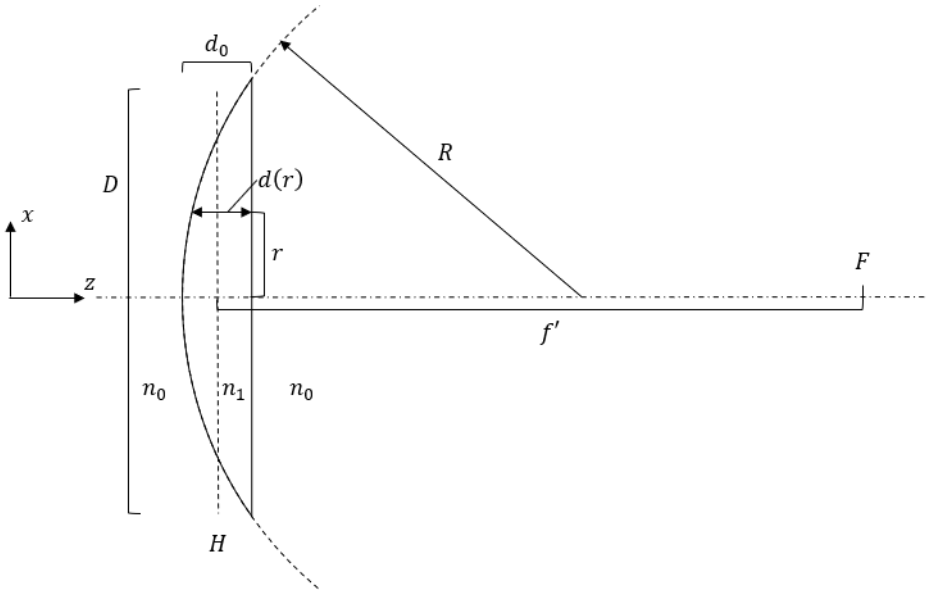


Figure 4-19: Two-dimensional sketch of a plano-convex lens in thin lens approximation with the refractive index n_1, aperture diameter D, apex thickness d_0 and focal length f'. The convex surface has the radius R.

To get the aperture function, the OPD must be expressed in terms of a relative phase shift of each wavefront component.

[25] The optical path difference is related to optical path length by the relative differences of local absolute path lengths.

Simplifying equation (4-86) with (4-92) yields:

$$\Delta\varphi = \frac{2\pi}{\lambda} OPD(x,y) \tag{4-93}$$

The aperture function of an ideal lens t_{lens} becomes:

$$t_{lens}(\mathrm{x},\mathrm{y}) = T(x,y) \cdot e^{-i\frac{2\pi}{\lambda}\Delta n \Delta d} \tag{4-94}$$

or in respect to the focal length f with equation (4-89) and (4-90):

$$t_{lens}(\mathrm{x},\mathrm{y}) = T(x,y) \cdot e^{-i\frac{2\pi}{\lambda} f' \Delta n^2 \sqrt{(f'\Delta n^2)-x^2-y^2}} \tag{4-95}$$

However, in practical applications it might be appropriate to confine the transmission $T(x,y)$ of the aperture function by a circular window. Most lenses are circular, so a $circ\left(\frac{D}{2}\right)$ can confine the aperture function to a lens diameter D, so that:

$$t_{lens}(\mathrm{x},\mathrm{y}) = e^{-i\Delta\varphi_{lens}} \cdot circ\left(\frac{D}{2}\right) \tag{4-96}$$

Where inside the circle of diameter D the transmission is $T = 1$, so no absorption occurs, and outside $T = 0$. An example of a lens modulation function is shown in Figure 4-20. Hereby a relatively large focal length of $f' = 100\,m$ is chosen, so that the phase modulation in Figure 4-20 a) is visible and not aliased. The phase function has the appearance of a chirp function with increasing frequency components at the edges. The truncation of the $circ$-function can be seen in Figure 4-20 b). Here the diameter of the calculated lens is $d = 100\,mm$ and the focal length f' is only valid for $\lambda = 632.8\,nm$. The modelling of lenses as aperture function allows to rebuild beam propagation paths with actual beam de-/ focusing, collimation or correction inside a digital model. This is useful for calculating optical convolutional neural networks.

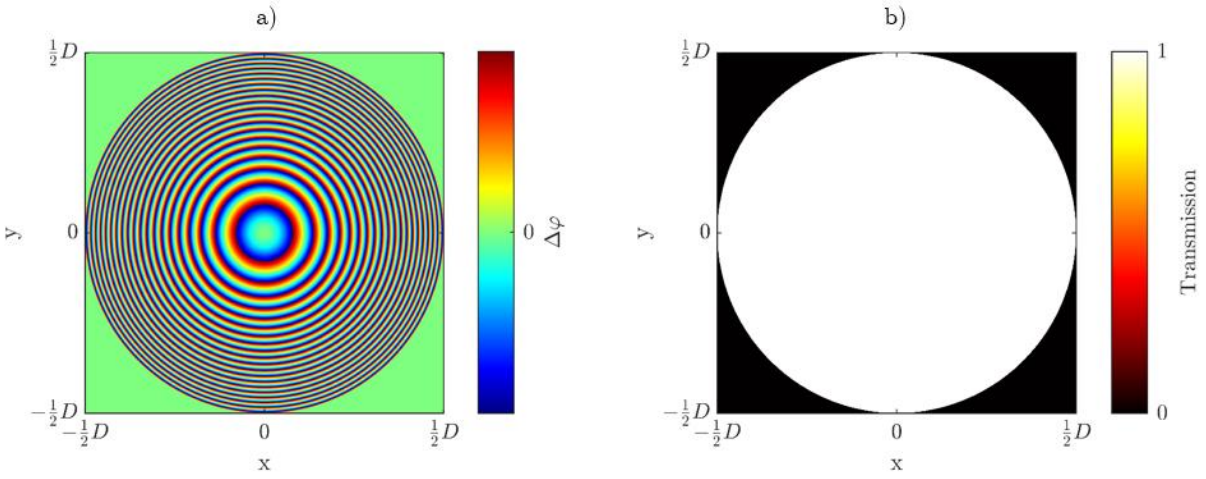


Figure 4-20: Modulation function of an ideal lens, with a diameter D of 100 mm, refractive index of $n_1 = 1.4$ and focal length $f' = 10^5\,mm$ for a wavelength of $\lambda = 632.8\,nm$. a) shows the phase angle of the modulation function and b) the amplitude truncated by the $circ\left(\frac{D}{2}\right)$ function.

4.4 BLAS algorithm for complex wave propagation in MATLAB

For calculating the in- and output of every layer of an optical neural network the BLAS (chapter 4.4) method is implemented using MATLAB. This chapter describes the algorithm for the propagation, calculation of the FSP function and a standalone algorithm for diffraction simulation.

4.4.1 Description of the Propagation Algorithm

The propagation calculation with the calculation of the FSP function is the essential function of the BLAS algorithm. An overview of the propagation algorithm is shown in Figure 4-22. One input is the complex wavefront amplitude u before the aperture plane in the form of:

$$u(\mathrm{x},\mathrm{y};\mathrm{z}_0^-) = A(x,y)\cdot e^{-i\left(k\sqrt{x^2+y^2}+\varphi(x,y)\right)} \tag{4-97}$$

Where z_0 is at the input plane and is always zero[26]. $A(x,y)$ is the real-valued amplitude and $\varphi(x,y)$ the phase at x and y. Although u is written as a continuous function of x and y, u is sampled in the interval Δx and Δy. A more accurate description of x and y would be x_n or y_m with the respective sample number n and m. However, under the premise that all functions are discrete data vectors, as well as the positional vectors x and y, the sample number subscript is omitted. The second input is the complex transmission function, a.k.a. aperture function $t(x,y)$, according to:

$$t(\mathrm{x},\mathrm{y}) = T(x,y)\cdot e^{-i\Delta\varphi(x,y)} \tag{4-98}$$

In equation (4-98) T is the respective local transmission coefficient of the modulating aperture and $\Delta\varphi$ is the local phase shift. To modulate the complex wave, every discrete element of the input wave amplitude $u(x,y;z_0^-)$ is multiplied with the element in the aperture function $t(x,y)$ respective of the position i.e., an element-wise multiplication. So that:

$$u(\mathrm{x},\mathrm{y};\mathrm{z}_0^+) = t(x,y)\circ u(x,y;z_0^-) \tag{4-99}$$

u at z_0^+ is the plane directly after the aperture. When the complex wave amplitude is modulated by equation (4-99), it has to be zero-padded before propagation according to the guidelines in chapter 4.2, to yield $u_P(x,y;z_0^+)$. The padding size must be determined beforehand so that the FSP function is calculated with respect to the larger computational window. The padding process is handled by the MATLAB function *padarray* [69]. For now, the padding process will

[26] All calculations are relative in terms of the calculation planes distances.

be represented by $\chi_P = pad\{\chi\}$, where χ is an arbitrary two-dimensional matrix and χ_P is double the size of χ. The angular spectrum of u_P is calculated by the means of the FFT (see chapter 4.2.1). So, the angular spectrum U_P is:

$$U_P(\nu_x, \nu_y; z_0) = FFT\{pad\{u(\mathrm{x}, \mathrm{y}; \mathrm{z}_0^+)\}\} \tag{4-100}$$

Again, the sampling subscripts of $\nu_x = \Delta\nu_x N_P$ and $\nu_y = \Delta\nu_y M_P$ are dropped. The $FFT\{\chi\}$ function assumes that the resulting spectrum is zero-shifted according to the method in chapter 4.2.1. The propagated angular spectrum is calculated by the element-wise multiplication of U_P with the FSP function H and the propagated field in the spatial domain $u_P(x, y; z)$ by taking the inverse transformation $IFFT\{U_P\}$ of U_P. The removal of the padding is crucial to remain a constant window size $N \times M$ of the input and output matrix and as stated in chapter 4.2.2, the padded region does contain aliased signal components. Let the vector $x_P = [x_1 \dots x_j]$ be the positional coordinates in x-direction and j be an integer running from 1 to N_P. When the total padding number is $N_P = N + P_x$, then $x = \left[x_{1+0.5P_x} \dots x_{N_P - 0.5P_x}\right]$ is the positional vector with the padding removed and with a length of N. The padded calculation window with the subscript convention is shown in Figure 4-21, where the matrix u is of size $N \times M$ and u_P of size $N_P \times M_P$, so that $P_x = N_P - N$ and $P_y = M_P - M$.

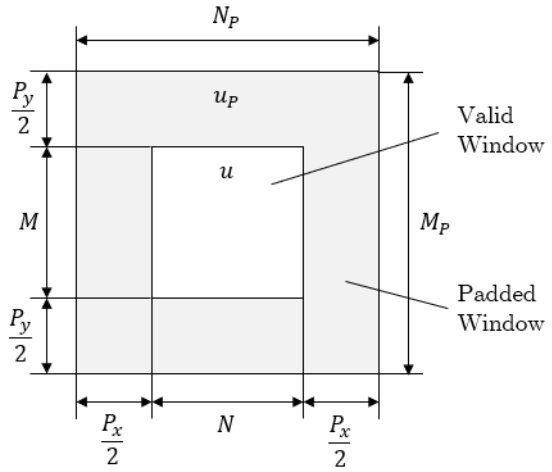


Figure 4-21: Illustration of the subscript convention of a padded calculation window u_P. Where u is the actual window, surrounded by P_x invalid data points in x-direction and P_y invalid data points in y-direction. The total size of u_P is $N_P \times M_P$.

The removal will be substituted by the function $\chi = pad^{-1}\{\chi\}$. So, the complete algorithm pictured in Figure 4-22 can be written as:

$$\boxed{u(x, y; z) = pad^{-1}\left\{IFFT\left\{FFT\left\{pad\{t(x, y) \circ u(\mathrm{x}, \mathrm{y}; \mathrm{z}_0^-)\}\right\} \circ H(\nu_x, \nu_y)\right\}\right\}} \tag{4-101}$$

The calculation of the FSP function H will be topic of the next chapter 4.4.2.

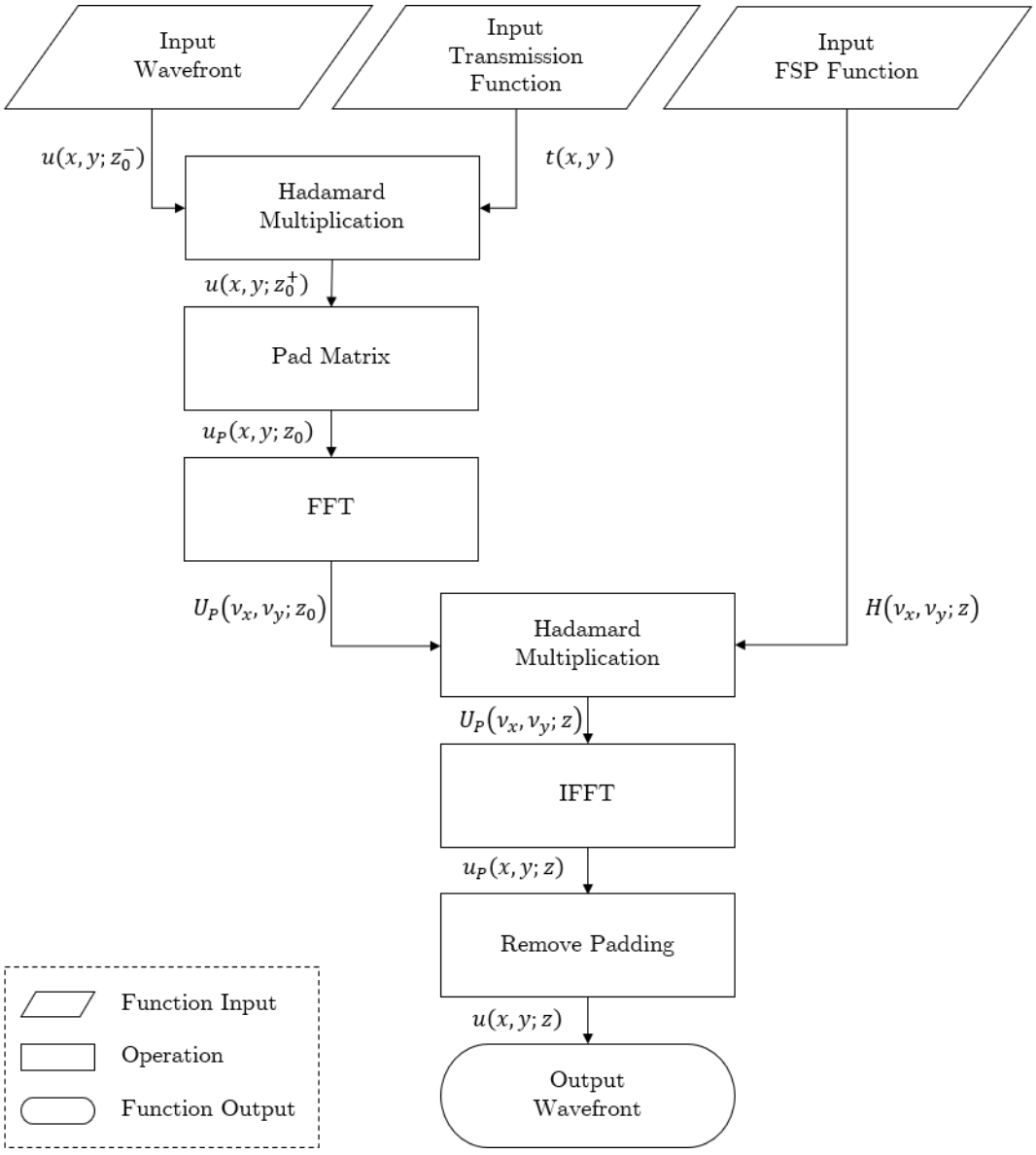


Figure 4-22: Flowchart of the BLAS/AS propagation algorithm.

4.4.2 Calculating the FSP Transfer Function

The FSP transfer function H is the matrix that relates an input amplitude spectrum $U(z_0)$ to an amplitude spectrum $U(z)$ at a distance z. The FSP transfer function H is an input of the BLAS propagation algorithm from chapter 4.4.1. H is depending on the computation window size, spatial sampling rate, wavelength, and the propagation distance. For an isotropic homogenous medium the FSP transfer function is a chirp function like shown in Figure 4-12, that scales in local spatial frequency relative to the propagation distance, computation window size and wavelength. To set up the dimensional relation the frequency sampling is computed by:

$$\Delta\nu_x = \frac{1}{N_P \Delta x} \; ; \; \Delta\nu_y = \frac{1}{M_P \Delta y} \tag{4-102}$$

With N_P, M_P and Δx, Δy being the number of sampling points of the padded window in x-/y-direction and spatial sampling distance, respectively. It is known from chapter 4.2.1 that the FFT-function produces an output with the same size as the input. So, the spectral input function U has the same size as u, therefore the spectral function that transforms every plane wave component in U also must have the size of $N_P \times M_P$. From this, the frequency vectors ν_x and ν_y are just:

$$\nu_x = n \cdot \Delta\nu_x \,, n \in \mathbb{Z} = \left[-\frac{N_P}{2} \dots \frac{N_P}{2}\right] \tag{4-103}$$

And:

$$\nu_y = m \cdot \Delta\nu_y \,, m \in \mathbb{Z} = \left[-\frac{M_P}{2} \dots \frac{M_P}{2}\right] \tag{4-104}$$

The scaling of the frequency axis already has been established in chapter 4.2.1, but to emphasize the importance of scaling when calculating the FSP transfer function, it is again repeated here. The frequency axis have the minimum and maximum values $\nu_{min,x} = -\frac{1}{2\Delta x}$ to $\nu_{max,x} = \frac{1}{2\Delta x}$ (analog in y-direction) and the frequency bin spacing of $\Delta\nu_x = \frac{1}{N_P \Delta x}$; $\Delta\nu_y = \frac{1}{M_P \Delta y}$. Now, with the spectral coordinates ν_x and ν_y the FSP transfer function can be calculated, according to (4-59), by:

$$H_0(\nu_x, \nu_y; z) = e^{i2\pi \cdot z \sqrt{\frac{1}{\lambda^2} - \nu_x^2 - \nu_y^2}} \tag{4-105}$$

The evanescent components are omitted by truncating H_0 with:

$$H(\nu_x, \nu_y; z) = H_0 \cdot circ\left(\lambda^{-2}\sqrt{\nu_x^2 + \nu_y^2}\right) \tag{4-106}$$

Where the $circ$-function is just a circular spectral low-pass filter with a radius of $r = \lambda^{-2}\sqrt{\nu_x^2 + \nu_y^2}$. The difference between the AS and the BLAS method is the application of an additional rectangular low-pass filter with the cut-off frequency at $\nu_{x,\max}$ and $\nu_{y,max}$ defined in equation (4-76) and (4-77) as:

$$\nu_{x,\max} = \frac{1}{\lambda\sqrt{4\Delta\nu_x^2 z^2 + 1}} \;\; ; \;\; \nu_{y,\max} = \frac{1}{\lambda\sqrt{4\Delta\nu_y^2 z^2 + 1}} \tag{4-107}$$

The spectral band limiting filter can be described by two rectangle functions and the FSP function of the BLAS method is:

$$H_{BLAS}(\nu_x, \nu_y; z) = H(\nu_x, \nu_y; z) \cdot rect\left(\frac{\nu_x}{2 \cdot \nu_{x,\max}}\right) rect\left(\frac{\nu_y}{2 \cdot \nu_{y,\max}}\right) \tag{4-108}$$

Where the actual width of rectangular filter is $2\nu_{x,\max}$ or $2\nu_{y,\max}$, because ν_x and ν_y run from $-\frac{1}{2\Delta x,y}$ to $\frac{1}{2\Delta x,y}$. The complete mathematical description of Figure 4-23 and the calculation of FSP transfer function is:

$$H_{BLAS}(\nu_x,\nu_y;z) = e^{i2\pi\cdot z\sqrt{\frac{1}{\lambda^2}-\nu_x^2-\nu_y^2}}\cdot circ\left(\lambda^{-2}\sqrt{\nu_x^2+\nu_y^2}\right)$$
$$\cdot rect\left(\frac{1}{2}\cdot\nu_x\lambda\sqrt{\frac{4}{N_P^2\Delta x^2}z^2+1}\right)rect\left(\frac{1}{2}\cdot\nu_y\lambda\sqrt{\frac{4}{M_P^2\Delta y^2}z^2+1}\right) \quad (4\text{-}109)$$

The equation (4-109) might look complex at a first glance, but in MATLAB these spectral filter can be created by logical operations and therefore the complexity of equation (4-109) reduces dramatically.

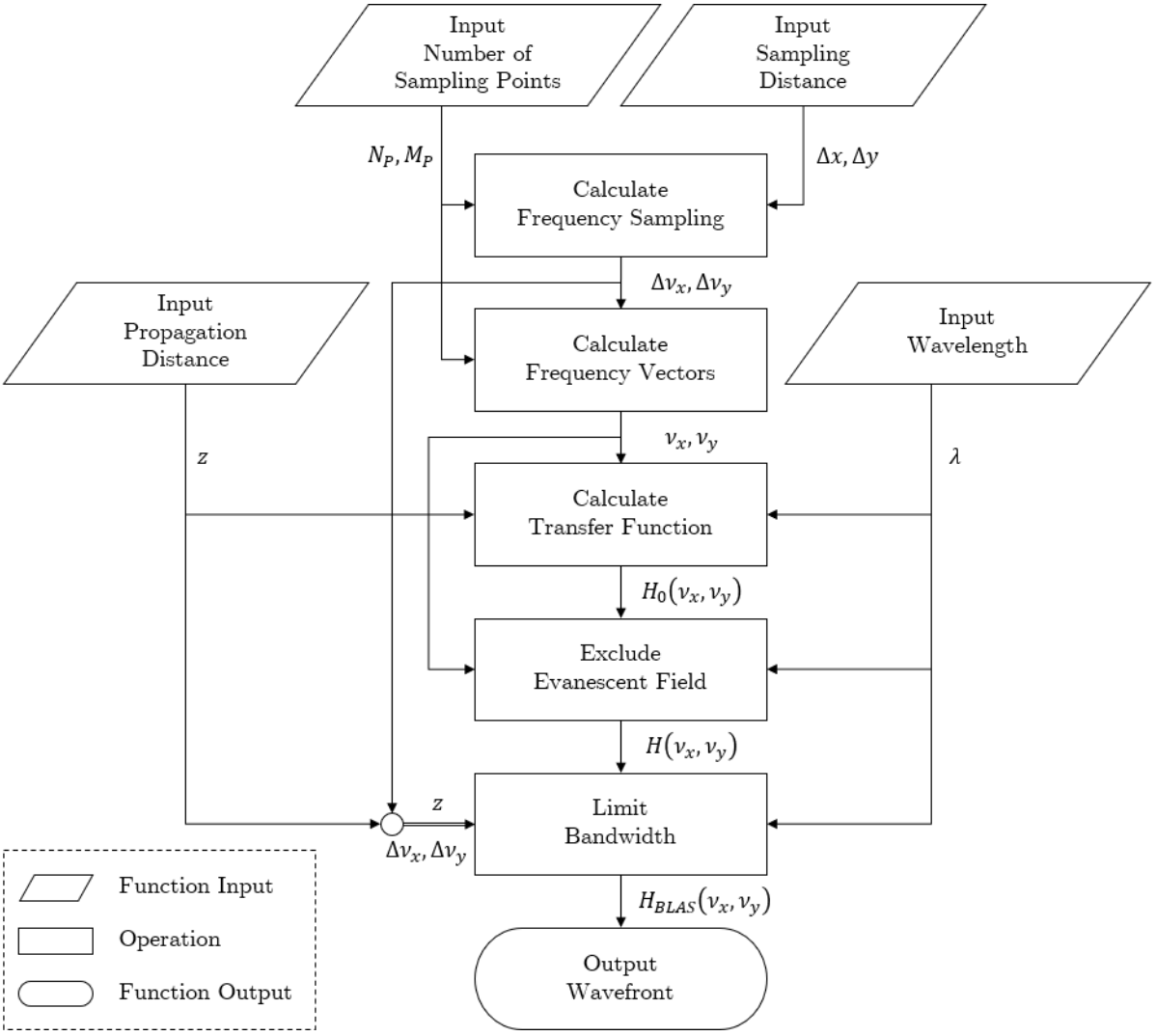


Figure 4-23: Flowchart of the algorithm for creating the FSP transfer function H.

4.5 Standalone Scalar Diffraction Simulation

The light propagation algorithm between two planes of chapter 4.2, the aperture modulation of chapter 4.3 and methods of chapter 4.4 might be also applied to other diffraction problems besides diffractive neural networks. More specifically, surface error evaluation and optimization, modelling diffractive optical systems or evaluating image quality deagradtion of optical components due to defects [70–72]. The algorithm described here is designed to take one or more arbitrary parallel complex transmission functions and a monochromatic optical wave as input and returns the resulting wave behind the system. The current state of this algorithm is limited to calculating modulation and propagation between parallel planes for monochromatic waves. The application is shown in Figure 4-24. An input wave is send throught an arbitrary number of optical elements. Those elements are modelled with their respective phase retardation (OPD) and relative transmission coeffiecients. The diffractive elements might also be thin lenses or other common optical elements. Using this algorithm, an arbitrary optical system can be predicted.

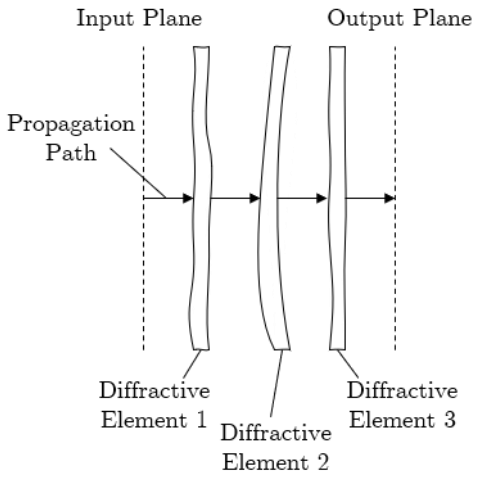


Figure 4-24: Principle of the satandalone wave propagation algorithm, where a field distribution gets repeatedly propagated and modulated. The field is evaluated at the output plane.

For further possible applications of this algorithm, like solving inverse problems and calculating reflecting surfaces, more information is are found in [Kapitel fehlt noch Outlook].

4.5.1 Definition of the Input Data

To ensure accurate simulation results the simulation paramters must be chosen with the guidelines given in chapter 4.2.3. The input parameters are:

- Wavelength λ:

  This paramter describes the wavelength inside the medium in which the wave propagates. The relative wavelength is $\lambda = \frac{\lambda_0}{n}$, where λ_0 is the wavelength in vacuum and n is the refractive index of the medium. Usually the refractive index is approximately $n \cong 1$ for air.

- Sampling interval Δs:

  The sampling interval defines the down- or upsampling of the input wave amplitude, the aperture function and the sampling of the output window. The sampling interval must be chosen to fulfill condition for Δs from chapter 4.2.3 but also has to sample the aperture function well enough. For the latter, it is hard to define a clear condition[27]. The sampling interval for one layer is a two-dimensional vector with the first entry being the sampling of the input wave and the second the sampling of the aperture.

- Computation window L:

  The computation window size defines the actual field size that is observed at the output and must be at least the size of the input field.

- Propagations distance z:

  The propagation distance is the distance of each plane to next.

An example for dimensioning a simulation setup is treated in detail for the experimental validation of the BLAS algorithm in chapter 5.3.1. All the above parameters are of the unit millimetres $[mm]$ and defined as an array with each entry defining the respective conditions of each layer.

The input wave front, that is to be propagated from the input to the output plane, is to be defined as a set of complex numbers $u_0(x, y; 0)$ that represents a sampled complex wavefront. Each matrix entry is calculated by:

$$u_0 = A(x, y) \cdot e^{-i\varphi(x,y)} \tag{4-110}$$

Where $A(x, y)$ is the mean amplitude and $\varphi(x, y)$ the relative phase at position x, y. Because of the statement in equation (4-17) and the prerequisite of a relative measurement, the amplitude might be related by the measurable value intensity by $A \sim \sqrt{I}$.

[27] The minimal sampling interval should be at least twice the highest sinusoidal frequency in the field to be sampled, according to Nyquist-Shannon [65].

The aperture function $t(x, y; l)$ is defined for each layer l in the same manner as the input wave amplitude, where the local transmission coefficient $T(x, y)$ is the amplitude of the complex function and the relative phase retardation is factor $\Delta\varphi(x, y)$. Each element of a layer becomes therefore:

$$t_l = T_l(x, y) \cdot e^{-i\Delta\varphi_l(x,y)} \tag{4-111}$$

It is important to mention that when modelling a physical objects' optical properties with a relative phase property or OPD for a specific wavelength λ, one must know the physical dimension and refractive index of that object, because the relative phase is defined in equation (4-86) as $\Delta\varphi(x, y) = \frac{2\pi}{\lambda} \cdot \Delta n(x, y) \cdot \Delta d(x, y)$. Usually, the property that can be measured is surface deviation and the refractive index is assumed to be homogenous in the medium.

4.5.2 Description of the Standalone Simulation Algorithm

A flowchart describing the steps taken by the standalone diffraction simulation algorithm is shown in Figure 4-25. For the calculation of a diffracted field through an arbitrary number of optical elements, the input complex wavefront and the first aperture are imported and then scaled according to the parameters given. The downsampling or upsampling factor SF is calculated by:

$$SF = \Delta s \cdot \frac{N}{L} = \frac{\Delta s}{\Delta \mathrm{l}} \tag{4-112}$$

Where Δs is the sampling distance defined as an input parameter, N is the number of data points in one direction[28], L is the physical length in one direction in units of millimeters $[mm]$ and Δl is the sampling interval of the input data. Equation (4-112) applies to the complex wave amplitude data u and aperture data t in the same manner. All data is scaled using the *imresize* function of MATLAB with bicubic interpolation [73]. The complex wavefront u is zero-padded or cropped, so that both have the same number of data points. The algorithm of chapter 4.4.2 calculates the complex transfer function. With the aperture data t_p, complex wave amplitude u_p and the FSP transfer function H the algorithm of chapter 4.5.1 calculates the propagated wavefront at the distance z^l. The output u^l is an intermediate result[29] that is saved into a variable in the MAT-file format [74] and will be used as the input into the next layer

[28] Assuming that, all data has the same length and number of elements in each direction.
[29] If the system consists of more than one layer and the calculated layer is not the last in the system.

calculation. If all layers are calculated the result is the last complex wave amplitude saved by the algorithm.

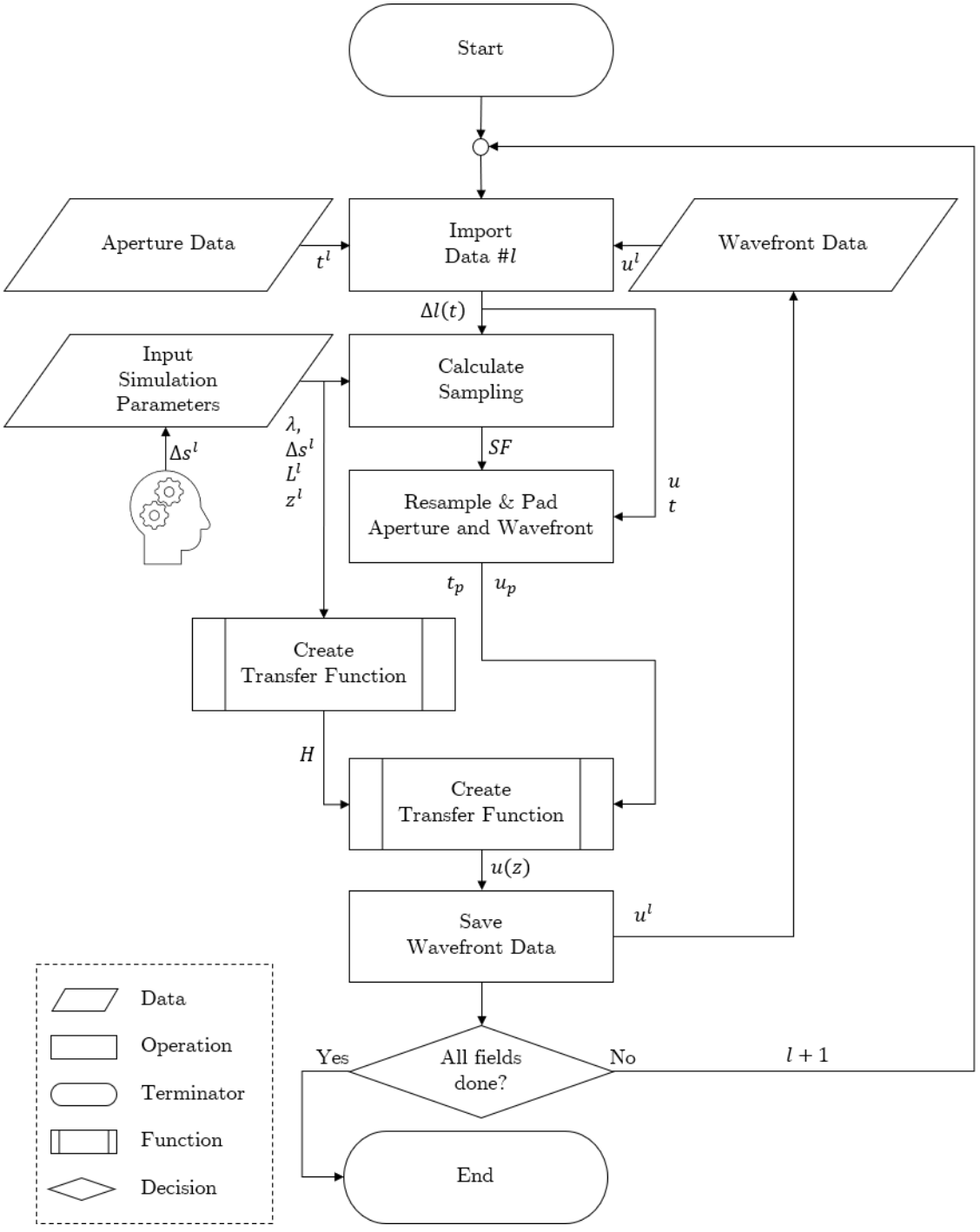


Figure 4-25: Flowchart of the standalone BLAS simulation algorithm.

5 Experimental

5.1 Computational Speed of the Bandlimited Angular Spectrum Method

The motivation for developing another method than the rigorous Rayleigh-Sommerfeld calculation for computing the diffraction pattern between adjacent diffractive neural network layers is the tremendous computational effort needed when using NIR or VIS wavelengths, according to Hypothesis 1. To get a measure of the advantage that the BLAS method has over a direct calculation of the RS integral, a speed comparison is made. For the RS integral to be accurate the sampling condition of [56] is obeyed. Several compuational aperture fields with increasing size serve as comparison. The cacluation of the matrix connecting the aperture field to the output field will not be viewed seperatly, because the size of the multiplication matrix in the RS integral method is too large to be calculated beforehand, because if N^2 is the number of sampling points, than the weight matrix would be the size N^4. In case of the calculation RS integral this is the matrix connecting each point in the aperture plane with each point in the image plane. In case of the direct integration method (DI) this is the free-space impulse response function h_{RS} and for the BLAS method this is the free-space transfer function H_{RS}.

Additionally, the allocated size of one compuational matrix is measured, as it gives a hardware limitation on the compuational field size and sampling. When only considering one propagation between two planes with one input optical field, one modulation, one transfer marix and one output optical field the total amount of matrices is four. In case of D^2NNs the network may consist of more than one layer and the computation may be needed to be parallelised to speed up the computation. To give the means for estimating the hardware resources necessary for a specific case, the size of one propagation element with four matrices is estimated by using the measurement of one matrix. Although the calculations might also been done on a graphical processor unit (GPU) or a solid state drive (SSD), do both methods have advantages and disadvanteges which limit their use only to specific cases. The disadavtantage of calculation on the GPU is that the dedicated memory of GPUs is typically limited to sizes of up to one magnitude smaller than the systems RAM. This limits the advantages of the GPU, which is large scale parallelism to matrix sizes optimized for the GPU memory size, and hinders the use of processor multithreating. If a network layout is highly optimized in terms of sampling, computation window size and so on, calculation on the GPU may offer significant advantages

in terms of calculation speed in exchange for flexibility. As for the calulation on a SSD, the case is quite the oppsite. A SSD offers large memory space but lacks the reading and writing speed of RAM and GPU memory. Because of this, is the SSD an alternative only when sampling conditions, calculation window size or network architecture require a memory size larger than can be offered by RAM. This might be the case for systems with a large number of computational threads, where the lack of reading and writing speed is compensated by the high degree of parallelism.

5.1.1 Experimental Setup

The first of the three methods compared is the rigorous calculation of the RS1 integral of equation (4-41) and the modification (4-55) which is in a discrete numerical sense derived as:

$$U_{RS1}(x,y;\Delta z) = \sum_{\xi,\eta} U(\xi,\eta;0)\cdot\frac{1}{r}e^{ikr}\left(\frac{1}{i\lambda}+\frac{1}{2\pi r}\right)\frac{z}{r}$$

Where x and y are the spatial coordinates of the output plane at distance Δz, ξ and η are the spatial coordinates of the input plane, the relative distance between each point is $r = \sqrt{(x-\xi)^2+(y-\eta)^2+\Delta z^2}$. The algorithm for the calculation of the RS integral is shown in Code 1.

Code 1: Rayleigh-Sommerfeld Integral calculation

```
% N: Number of samples in each direction
% x: Posititon vector in with N elements
% z: Propagation distance scalar
% lambda: calculation wavelength
% k: Wavenumber 2.*pi/lambda
% U_0: Complex aperture field distrubution
% U: Initialized output field matrix

% Loop iterating through n columns & m rows
for n = 1:N
    for m = 1:N
        % Calculate respective position
        X = x-x(n);
        Y = (x-x(m))';
        % Calculate radial distance matrix
        r = sqrt(X.^2 + Y.^2 + z^2);
        % Rayleigh-Sommerfeld integral
        U = U + ...
            (U_0(n,m)...
            .* ((1/(1i*lambda)) + (1./(2*pi.*r))...
            .* z.*exp(1i*k.*r)./(r.^2)));
    end
end
```

The second method is the DI method, which is mathematically similar to the first method but uses a convolution to calculate the output field. The convolution kernel is the free-space impulse response function defined by equation (4-56). The output field is then defined as:

$$U_{DI}(\Delta z) = U(0) * h_{RS} = U(0) * \left(\frac{1}{r} e^{ikr} \left(\frac{1}{i\lambda} + \frac{1}{2\pi r}\right)\frac{z}{r}\right)$$

Hereby $r = \sqrt{x_i^2 + y_i^2 + \Delta z^2}$ and x_i, y_i are spatial coordinates for the impulse response function, which must be double the size of the calculation window and zero centered. The MATLAB code for the DI method is shown in Code 2.

Code 2: Direct Integration method impementation

```
% N: Number of samples in each direction
% sA: Size of the aperture
% ds: sampling distance of the aperture
% z: Propagation distance scalar
% lambda: calculation wavelength
% k: Wavenumber 2.*pi/lambda
% U_0: Complex aperture field distrubution
% U: Initialized output field matrix

x = linspace(-sA,sA-ds,2*N);
r = sqrt(x.^2 + x'.^2 + z^2);
h = (1/(1i*lambda)) + (1./(2*pi.*r)) .* z.*exp(1i*k.*r)./(r.^2);
U = conv2(U_0,h,'same');
```

The third method compared will be the BLAS method of equation (4-60) with a bandwidth limit defined by equation (4-80). The output field is then calculated by:

$$U_{BLAS} = IFFT\{H_{RS} \circ W_{max} \circ FFT\{U(0)\}\}$$

Whereby H_{RS} is free-space propagation function and W_{max} is the bandwidth limit.

Code 3: BLAS method implementation

```
% N: Number of samples in each direction
% du: frequency sampling distance 1/(2*aperture size)
% u: frequency coordinate vector with 2*N samples
% z: Propagation distance scalar
% lambda: calculation wavelength
% k: Wavenumber 2.*pi/lambda
% U_0: Complex aperture field distrubution

% Calculate free-space propagation transfer function
H = exp(1i*2*pi*z*sqrt(lambda^(-2) - u.^2 - (u'.^2)));
% Calculate bandwidth limit
ulim = ((2*du*z)^2 +1)^(-0.5)/lambda;
% Apply bandwidth filter
H = H .* ((abs(u)<=ulim)' * (abs(u)<=ulim));
% Add zero-padding to field
U_0_P = padarray(doe,[N/2 N/2],'both');
```

```
% Calculate frequency spectrum
U_0_F = fftshift(fft2(U_0_P));
% Free up RAM
clear U_0_P;
% Calculate output field
U = ifft2(U_0_F .* H);
% Free up RAM
clear U_0_F;
% Extract valid area
U = U((N/2)+1:1.5*N,(N/2)+1:1.5*N);
```

All three methods are calculated for a distance of $\Delta z = 10\,mm$. The sampling distance of the aperture field is $\Delta s = \lambda/2$ to match the highest needed sampling rate.

All calculations are performed using only the CPU, which is an AMD Ryzen 7 2700X with 8 cores and 16 logical processors at 3.7 GHz clock frequency. The total available RAM is 49 Gb DDR 4.

To show the limitations of the BLAS method, the allocation of physical memory of one matrix size is measured for single precision complex values and for double precision complex values. The theoretical sizes of those two precision types are defined by the IEEE Standard 754 [75]. According to this standard a double float precision data type has values in the range of approximately $2.2 \cdot 10^{-308}$ to $1.8 \cdot 10^{308}$ for positive numbers and $-1.8 \cdot 10^{-308}$ to $-2.2 \cdot 10^{308}$ for negative numbers with around 15 decimal places. A single float precision data type contains values in the range of approximately $-3.4 \cdot 10^{-38}$ to $-1.2 \cdot 10^{38}$ for negative numbers and $1.2 \cdot 10^{-38}$ to $3.4 \cdot 10^{38}$ for positive numbers with around 7 decimal places. In terms of size, a complex double precision value allocates $16\,bytes$ and a complex single precision value exactly half which are $8\,bytes$. A complex number therefore holds two sperate values, one value for the real part and one value for the imaginary part.

Note that the impact of precision on the simulation values has not been researched in this work. It is assumed that no overflow nor underflow occurs.

With the theoretical value of complex single and double precision values defined in [75], a prediction of the matrix sizes can be made. Let N^2 be the number of matrix elements, then the matrix size for single and double precision complex numbers becomes $N^2 \cdot 8\,bytes$ and $N^2 \cdot 16\,bytes$ respectively. The actual size is measured in this experiment to validate this prediction. A possible overhead may change the actual memory allocation.

5.1.2 Results

The results of the first part of the calculation time measurement are shown in Figure 5-1. Each data point is the mean value of three measurements. The horizontal axis represents the size of a quadratic matrix in one direction N, so that the total number of sampling points is N^2. Note that the scale of the vertical axis is logarithmic. Also note that the calculation of the free-space impulse response function for the DI method and the free-space transfer function are included in the time measurements. If an optical wave field would be propagated through several planes with equal distance, these functions only must be calculated once. However, the propagation distance of $\Delta z = 10\,mm$ is relatively small, which results in minimal padding of factor two for the calculation window for the DI and BLAS methods. This zero-padding operation is also included in the calculation time measurements. The fitted curves on the measured data are power functions in the form of $y(x) = a \cdot x^b$. The coefficients a and b for the Rayleigh-Sommerfeld (RS) calculation are $a = 1.448 \cdot 10^{-4}$, $b = 2.089$, for the direct integration method (DI) $a = 9.372 \cdot 10^{-11}$, $b = 3.845$ and for the band limited angular spectrum (BLAS) method $a = 9.067 \cdot 10^{-8}$, $b = 2.051$. The first measurement points of the DI and BLAS method in Figure 5-1 do not match the fitting type. The convolution either in spatial or spectral coordinates for smaller window sizes do not scale according to the fitting curve, like larger matrix sizes do, due to the presents of overhead calculations. The precision used are single precision complex numbers.

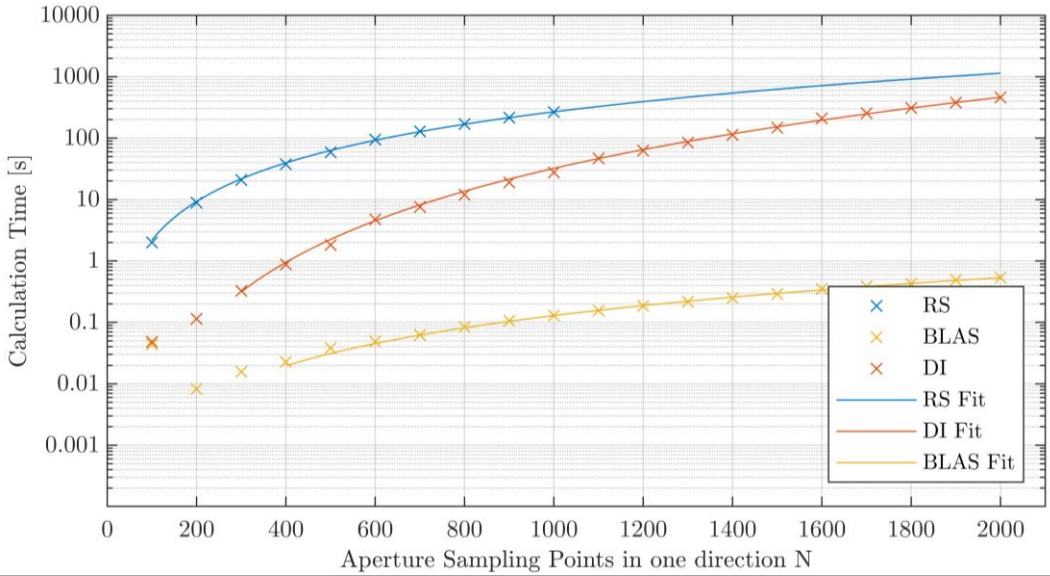


Figure 5-1: Calculation speed measurement of three diffraction calculation methods. The methods are the rigorous calculation of the Rayleigh-Sommerfeld integral (RS), the convolution-based direct integration method (DI) and the band limited angular spectrum method (BLAS).

A further investigating is shown in Figure 5-2 of the computation of the BLAS method in terms of the contribution of each calculation type, that are calculation of the transfer function, the calculation of the propagated field with the computation of the FFT and IFFT of the field, and the multiplication of the optical field with the diffractive layer i.e., the field modulation. The calculation of the transfer Function H_{BLAS} takes approximately a third of the complete calculation time. The FFT and IFFT of the optical field are the main contribution to the complete calculation time, as the multiplication of two complex field takes only approximately 3-4 % of the time.

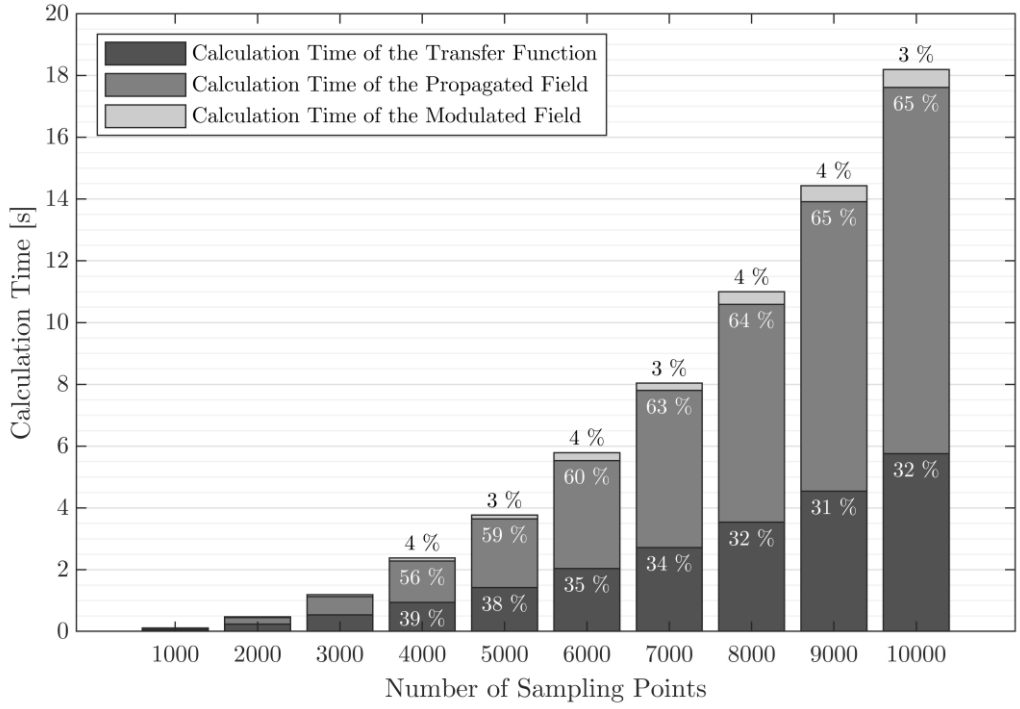


Figure 5-2: Composition of computation steps of the BLAS method as stacked bars. The percentages at each element denote the contribution of each calcuation to the overall calculation time.

Isolating the optical field propagation from the computation of the transfer function yields the calculation times shown in Figure 5-3. The fitted curve is again a power function in the form $y(x) = a \cdot x^b$, where the coefficients are $a = 4.026 \cdot 10^{-9}$ and $b = 2.373$ with a R^2 value of 0.9997.

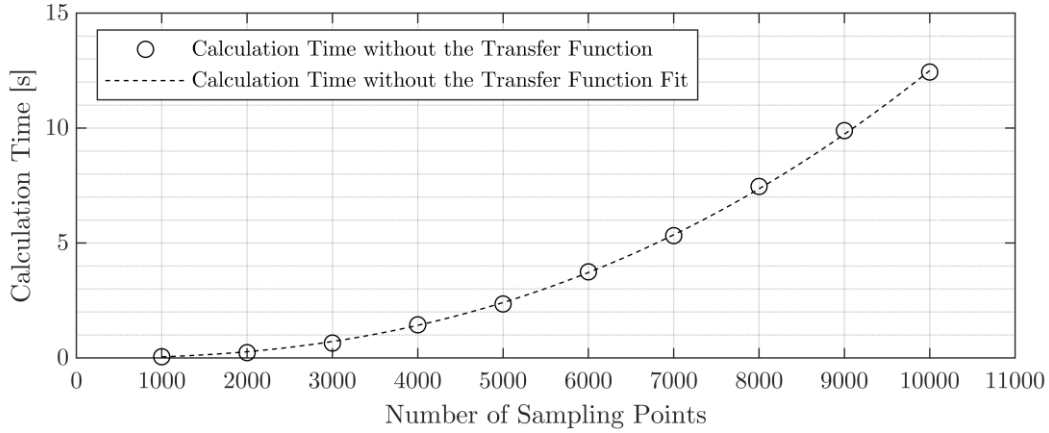


Figure 5-3: Isolated calculation times for the computation of the optical field propagation of the BLAS method. The functions used are one FFT, a element-wise matrix multiplication and a IFFT. The calculation time is measured for several Matrix samplings. The total number of matrix elemnts is the square of the sampling points.

The results for the measurement of the physical memory allocation on the systems' RAM is shown in Figure 5-4, whereby the change in available RAM size is measured using the *memory* [76] function of MATLAB. The continuous and dashed lines are the calculated values of single and double precision complex matrices, respectively. The measured sizes correspond to the expectation by the theoretical sizes.

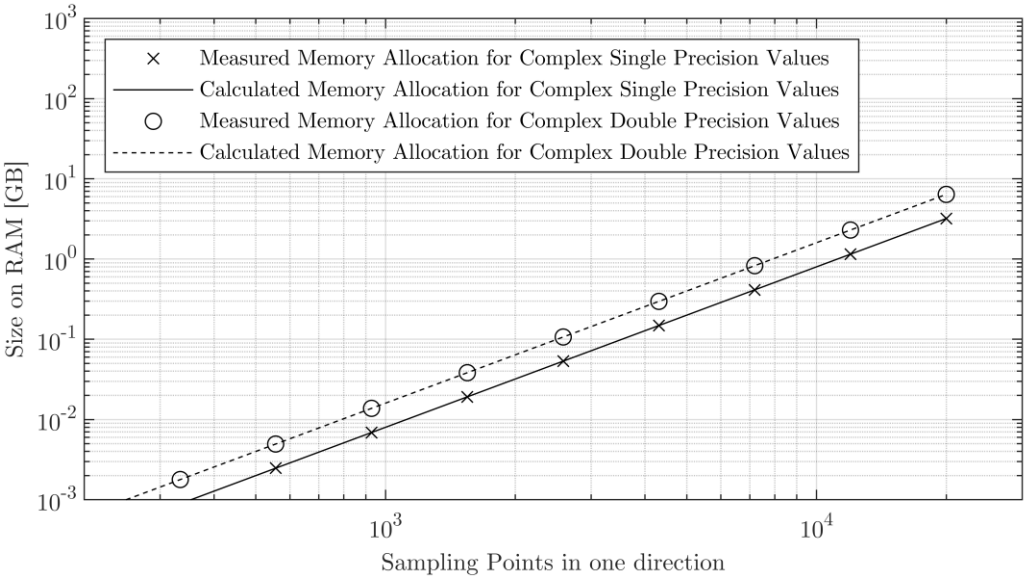


Figure 5-4: Measurement of the RAM usage by increasing sizes of single and double precision complex matrices. Whereby the horizontal axis is the number of points in one direction, the actual matrix size would be the value squared.

5.1.3 Discussion

The results of Figure 5-1 show the advantages of the BLAS method developed in section 4.2 in terms of calculation speed, which is magnitudes faster than the rigorous RS calculation or the

convolution-based direct integration method. Furthermore, the necessary sampling for the rigorous RS calculation differs from convolution-based methods. Figure 5-5 shows a comparison between the sampling condition of the RS method based on equation (3-17) and the BLAS conditions derived in subsection 4.2.3. The maximum sampling distance is calculated for a unitary aperture size of $a = 1$. Note that the sampling depends on the aperture size in both cases. The plot shows that with increasing propagation distance the allowable sampling distance increases for the BLAS method, which in turn decreases the absolute size of calculation matrices and increases the calculation speed. The bandwidth limit of the BLAS method ensures correct field calculation if the zero-padding is sufficient.

In addition to the absolute advantages in computational speed, the decrease of necessary sampling further benefits the computational efficiency. This makes the BLAS method especially suited for the calculation of the propagation between D^2NN layers. The sampling conditions derived in subsection 4.2.3 give a lower limit for sampling each. Based on the ratio of spatial sampling of the aperture function and the neuron density it is correct to assume that each neuron is typically oversampled when computing with optical wavelengths and structures multiple times larger than the operating wavelength. Oversampling is useful for including surface derivations of layers and system noise[30].

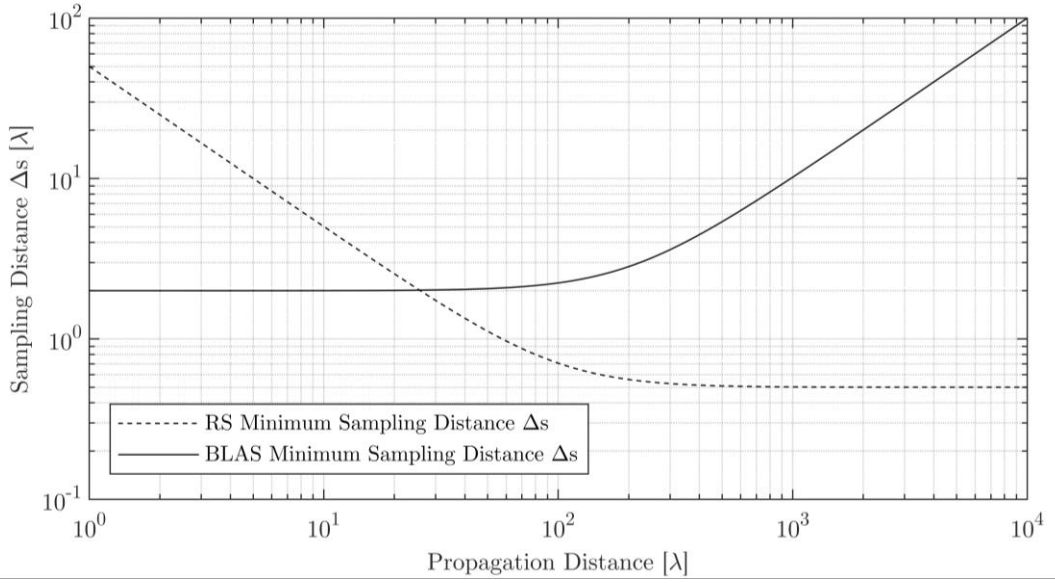


Figure 5-5: Calculated sampling conditions for the RS (dashed) and BLAS (continous) method in wavelengths $[\lambda]$ *for a unitary aperture size of* 1.

[30] See the Outlook chapter 7 for a more detailed hypothesis on surface oversampling.

For example, a network layer with 200×200 neurons may have a layer size of $5 \times 5\ mm^2$. The size of one single neuron therefore becomes $\frac{5}{200}\ mm = 2.5\ \mu m$. The layer distance is chosen to be $\Delta z = 10\ mm$ and the operation wavelength is $\lambda = 633\ nm$. Then, by using the sampling conditions of subsection 4.2.3, the required spatial sampling becomes:

$$\Delta s = 2\lambda\sqrt{4\Delta v^2 z^2 + 1} = 2\lambda\sqrt{\frac{4}{L^2}z^2 + 1} = 2 \cdot 633\ nm\sqrt{\frac{4}{5^2\ mm^2}10^2\ mm^2 + 1} \cong 5.2\ \mu m$$

This sampling distance of $5.2\ \mu m$ results in a matrix size of $\left(\frac{2 \cdot 5\ mm}{5.2\ \mu m}\right)^2 \cong 1924^2$ elements, whereby the zero-padding is already considered. Comparing Δs to Figure 5-5, reveals that with increasing aperture size the needed sampling distance decreases[31]. In this calculation example, the sampling distance is $\Delta s \cong 8.2\ \lambda$. The sampling distance in the RS calculation case would be $\Delta s = 0.6\ \lambda$[32], which is approximately 14 times smaller. Using the results of Figure 5-1 to estimate the computation time $t_{l=1}$ for the propagation calculation yields:

$$t_{l=1} = 9.067 \cdot 10^{-8}\ s \cdot N^{2.051} \cong 0.49\ s$$

Whereby $N = 1924$ is the number of sampling points in one direction with zero-padding. The calculated time of $t_{l=0} = 0.49\ s$ includes the calculation of the transfer function H_{BLAS}. In the case of several layers with equidistant spacing the transfer function must be calculated not only once per layer pass but once per complete D^2NN network training. This reduces the calculation time dramatically. Using the results of Figure 5-3 the calculation time of additional layers can be derived as:

$$t_l = 4.026 \cdot 10^{-9}\ s \cdot N^{2.373} \cong 0.25\ s$$

For exemplary purposes, the fictive network may consist of five layers. Then, the number of propagations becomes six. The number of propagations is always one more than layers in the network because the distance from the last layer to the screen also must be considered as shown in Figure 5-6. Note that the use of activation functions is not considered here, because as with the multiplication of modulation the computational effort is minimal.

[31] The calculated sampling distance does not correspond to Figure 5-15, which is calculated for a circular aperture of $1\ mm$. Increasing the aperture effectively shift the curve of Figure 5-15 to the right.

[32] The distance to the edges of the squared window is the highest radial distance from the image center. Therefore, the aperture size is $\sqrt{2} \cdot 10\ mm$.

However, a D²NN with five layers then has a computation time for the forward pass of $t_{fw} = t_{l=1} + 5t_l = 0.49\,s + 5 \cdot 0.25\,s = 1.74\,s$. Here the conjecture is made that the backpropagation of the network error through the network also must be calculated using the BLAS method, but this should be subject to further investigations on a network prototype itself. By using this conjecture, the calculation time must be doubled. Furthermore, when using the MNIST hand-written number database containing 60,000 samples, one epoch of training would take:

$$t_{epoch} = 60{,}000 \cdot 2 \cdot t_{fw} = 58\,h$$

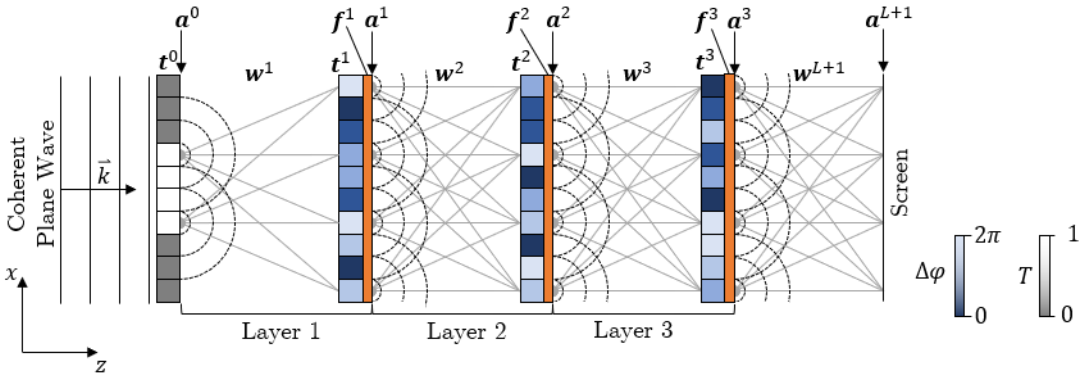


Figure 5-6: Forward Propagation of a D²NN with three diffractive layers according to [9] and [48]. An incoming monochromatic plane wave is modulated by $\boldsymbol{t}^0$ in amplitude to form the input image. Through several layers of propagation $\boldsymbol{w}^l$, modulation $\boldsymbol{t}^l$ and activation $\boldsymbol{f}^l$ the diffracted optical wave $\boldsymbol{a}^{L+1}$ falls onto an observation screen.

In contrary, the calculation time for the RS method would become $17.51\,min$ per layer, which would yield a computation time for one epoch of $t_{epoch} = 60000 \cdot 2 \cdot 6 \cdot 17.51\,min \cong 24\,years$. This large computation time might be reduced by calculating the radial distances once. The radial distance matrix, the input and the resulting matrix must then be kept in memory. For this example, the radial distance matrix becomes the size of $1924^4 \cdot 8\,bytes \cong 109.6\,Tb$ for single precision numbers, which not feasible at all.

Especially this comparison for a typical scenario shows that precise diffraction calculation of optical fields in the VIS and NIR range for D²NN is only possible by using the BLAS algorithm with the derived sampling conditions. Further, the BLAS algorithm provides the means for calculating the field propagation with an oversampling factor for noise and surface deviation inclusion. Also note that the CPU used for time measurement is a basic commercial processor unit.

To check, if the matrices for one calculation fit the RAM, the memory usage is calculated. The memory allocation for a single precision matrix is according to the results of Figure 5-4:

$$N^2 \cdot 8\,bytes = 1924^2 \cdot 8\,bytes \cong 29.6Mb$$

For a five-layer network one input, one output, five transmission, five forward optical field and five backpropagation field matrices must be kept in the memory at the same time preferably[33]. The total needed memory for calculating one sample therefore becomes:

$$17 \cdot 29.6\,Mb = 503.4\,Mb$$

As typical RAM sizes are $\geq 16\,GB$, this is not a critical point in this example. Furthermore, all calculation matrices fit on a typical GPU memory and might be calculated on the GPU. Note that all calculations in this example do not account for any overhead due to network training, like graphical output parameters updates and so on.

5.2 Scalar Diffraction Simulation of a Convolution Unit

For the verification of Hypothesis 3 the BLAS method is used to model multiple optical imaging systems. This experiment is structured as follows: In the first part a single imaging lens and in the second part a diffraction grating is modelled using the method from section 4.3. The results are compared with the analytical expectations derived from ray optics. In the third part a 4-f system consisting of two imaging lenses is analyzed. The image in the Fourier plane is compared with a Fourier transformation of the image. In the fourth part a complete convolutional setup with a lens array is investigated.

For all simulations in this section the in subsection 4.4.1 and 4.4.2 presented algorithms are used for single-propagation systems and for multiple propagations the addition of the algorithm in section 4.5 is used. The MATLAB functions written for this section are given in the following. Hereby the function *createLensAperture* generates a spherical lens with a given focal length and diameter shown in Code 4, *createTransferFunction* generates the spectral free-space transfer function H_{BLAS} based on the propagation distance shown in Code 5 and *propagateField* calculates the resulting optical field at the observation or intermediate plane shown in Code 6.

[33] The intermediate results might temporarily be saved on a SSD drive as well, if the RAM size limits the maximum matrix size.

Code 4: MATLAB function for creating a thin lens model.

```
function [lens] = createLensAperture(...
    focalLength,...
    lambda,...
    refIndex,...
    diameter,...
    x)
%Create thin lens model based on a phase offset due to lens thickness and
%refractive index
%   Input parameters:
%       focalLength: Target focal length of the lens as a positive value
%       lambda: Operation wavelength
%       refIndex: Refractive index of the lens material
%       diameter: Lens aperture diameter
%       x:  positional vector, i.e. one-dimensional distance from the
%           optical axis, with the total size of the calculation window

    % Calculate the wavenumber
    k = 2*pi/lambda;
    % Calculate the change of refractive index
    dn = refIndex-1;
    % Calculate thins lens OPD
    lens = exp(...
        1i*k*focalLength*dn^2.*...
        sqrt((focalLength*dn^2)-x.^2-x'.^2));
    % Calculated the radial position vector
    r = sqrt(x.^2+x'.^2);
    % Limit lens by aperture radius
    lens = lens.*(r<=(diameter/2));
end
```

Code 5: Creation of the free-space transfer function for calculation of the propagated optical field.

```
function [H] = createTransferFunction(sField,nSamples,z,lambda)
% Creates a free-space transfer function
%   Input parameters:
%       sField: Physical size of the aperture in one direction
%       nSamples: Number of spatial samples in one direction
%       z: Propagation distance
%       lambda: Operation wavelength

    % Calculate spatial sampling distance
    ds = sField/nSamples;
    % Calculate spectral sampling distance
    du = 1/(2*sField);
    % Calculate bandwidth limit
    ulim = ((2*du*z)^2 +1)^(-0.5)/lambda;
    % Calculate the spectral coordinate vector
    u = linspace(-(1/(2*ds)),(1/(2*ds)),nSamples);
    % Check if bandwidth limit truncates more than 90% of the spectrum if not
    % increase zero-padding size until 10 % of the spectrum is left
    % after limiting the bandwidth
    overPad = 1;
    while sum((abs(u)<=ulim),'all') < ((nSamples/10))
        % Increase overpadding by 1
        overPad = overPad + 1;
        % Check if samplin number is an integer
        if mod(overPad*nSamples,1) == 0
            % Update values
            sField = overPad*sField;
            nSamples = overPad*nSamples;
            du = 1/(2*sField);
            ulim = ((2*du*z)^2 +1)^(-0.5)/lambda;
```

```
            u = linspace(-(1/(2*ds)), (1/(2*ds)), nSamples);
        end
    end
    % Calculate free-space propagation function
    H = exp(1i*2*pi*z*sqrt((1/(lambda^2))-u.^2-u'.^2));
    % Apply bandwidth limit
    H = H .* ((abs(u)<=ulim)' * (abs(u)<=ulim));
end
```

Code 6: MATLAB function for the calculation of the propagated optical diffraction field

```
function [outputField] = propagateField(inputField,H)
%Calculates the FFT-based convolution of the input field and the transfer
%function.
%   Input parameters:
%       inputField: Optical Field in the spatial domain
%       H:  Free-space transfer function calculated by
%           createTransferFunction().
%   Note: inputField and H must be square matrices of the same size.

    % Calculate the spectrum of the input Field
    inputSpectrum = fft2(inputField);
    % Zero-shift the transfer function to match the frequencies of the
    % input spectrum
    H = fftshift(H);
    % Calculate the output taking the inverse FFT of the elementwise
    % multiplied fields.
    outputField = ifft2(H.*inputSpectrum, 'nonsymmetric');
end
```

5.2.1 Experimental Setup Part 1

The setup for the simulation of the single lens is shown in Figure 5-7. A plane wavefront with constant energy is modulated by lens aperture. The lens is modelled by the equations (4-95) and (4-96) as follows:

$$t_{lens}(\mathrm{x},\mathrm{y}) = circ\left(\frac{D}{2}\right) \cdot e^{-i\frac{2\pi}{\lambda} f' \Delta n^2 \sqrt{(f'\Delta n^2)-x^2-y^2}}$$

So, to measure if the target focal length f' is correct, a lens diameter of $D = 8\,mm$ was chosen. The calculation wavelength was $\lambda = 633\,nm$. The refractive index change was chosen to be $\Delta n = n_{lens} - n_{air} = 1.4 - 1 = 0.4$. The focal length of the lens was designed for $f' = 200\,mm$. The setup is sketched in Figure 5-7. The maximum intensity in the observation plane corresponds to the focal point. If this intensity maximum is found at the distance f', then the lens is assumed to be modelled correctly. The observation plane is sampled with 300 points and has a total length in z-direction of $2f' = 400\,mm$. The sampling of the aperture is $\Delta s = 10\lambda$. Each two-dimensional image was calculated completely, and a cross-section taken out at $y = 0$ to visualize the focal spot.

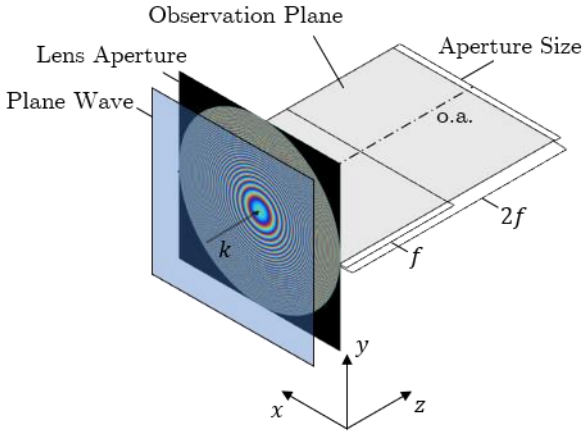


Figure 5-7: Setup for measuring the focal length of a modelled lens aperture illuminated by a monochromatic plane wave. The observation plane is in the x-z-plane and chosen so that the focal point ideally is in the observation plane center. The optical axis is indicated by o.a. and the direction of the incoming wavefront by k.

Secondly, at the focal plane an airy pattern with a central airy disc is predicted. Therefore, the image in the focal plane will be investigated, as illustrated in Figure 5-8. It can be shown that, the first minimum of the airy pattern can be found at a distance $d_{airy} \cong 1.22 \frac{\lambda f'}{D}$ from the optical axis. The setup for this test is a lens with a focal length of $f' = 20\,mm$, the refractive index change from air to lens is kept at $\Delta n = 0.4$, the lens and aperture diameter are $D = 1\,mm$. The sampling of the aperture is $\Delta s = \lambda/2$. The observation plane is in the focal plane of the lens. The expectation for this setup regarding the first minimum of the airy pattern is that it can be found at a distance $d_{airy} \cong 1.22 \frac{633\,nm \cdot 20\,mm}{1\,mm} = 15.4\,\mu m$.

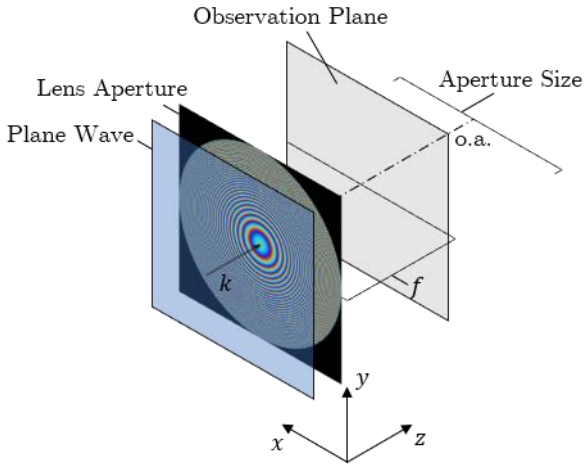


Figure 5-8: Setup for observing an airy pattern created by a plane wave passing a circular aperture and a modelled lens. The observation plane is located at the focal plane of the lens.

All results are shown in the next subsection in their respective order explained in this subsection.

5.2.2 Results Part 1

The measurement of the cross-section for evaluating the focal length is shown in Figure 5-9. The maximum intensity is at exactly $200\,mm$ from the aperture, which correspond to the expected focal length f' of the simulated lens. Some negative spherical aberration can be observed around the focal spot as converging lines.

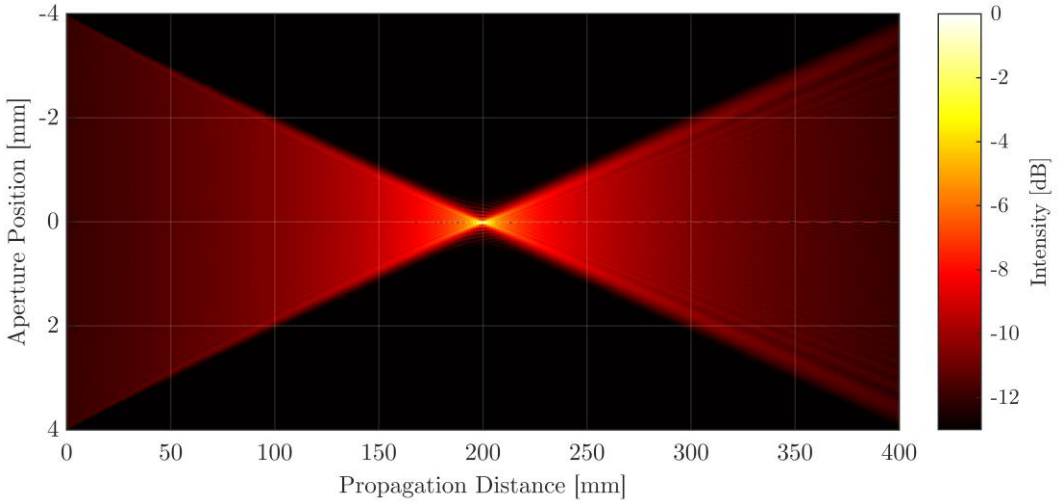


Figure 5-9: Cross-section along the x-z-plane of a simulated monochromatic focused wavefront by a modelled lens with a focal length of $f' = 200\,mm$. The aperture diameter is $D = 8\,mm$ and the wavelength is $\lambda = 633\,nm$. The shown intensity is in a logarithmic scale of normalized intensity.

The simulation of the second setup, where the intensity in the focal plane is observed, is shown in Figure 5-10. The image shows a section around the optical axis. The intensity is in a linear scale. A slight halo around the central focal spot might be seen, which would correspond to the first ring of the airy pattern.

For a more precise view, a cross-section at the center of Figure 5-10 is shown in Figure 5-11. A typical Bessel function as airy pattern can be observed [54, pp 98–100]. The first minimum according to $d_{airy} \cong 1.22\frac{\lambda f'}{D}$ is drawn as a dashed line. The minimum of the intensity and the calculated minimum at $d_{airy} \cong 15.4\,\mu m$ do qualitatively correspond. However, the intensity profile is slightly decentered.

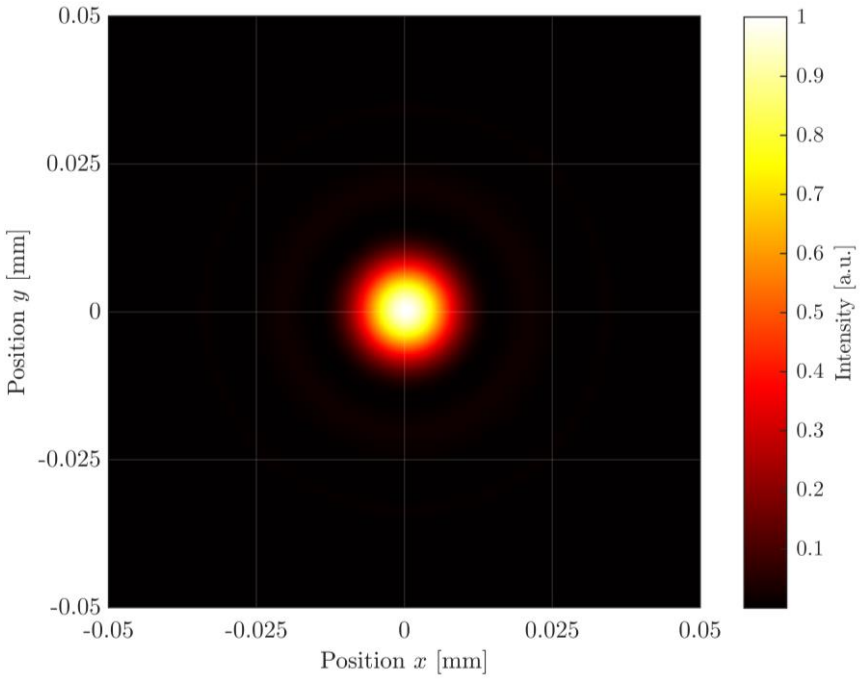


Figure 5-10: Intensity in the x-y-plane of a simulated focal spot in the focal plane. The aperture size is $D = 1\,mm$, the focal length of the lens is $f' = 20\,mm$, the wavelength is $\lambda = 633\,nm$ and the spatial sampling is $\Delta s = \lambda/2$.

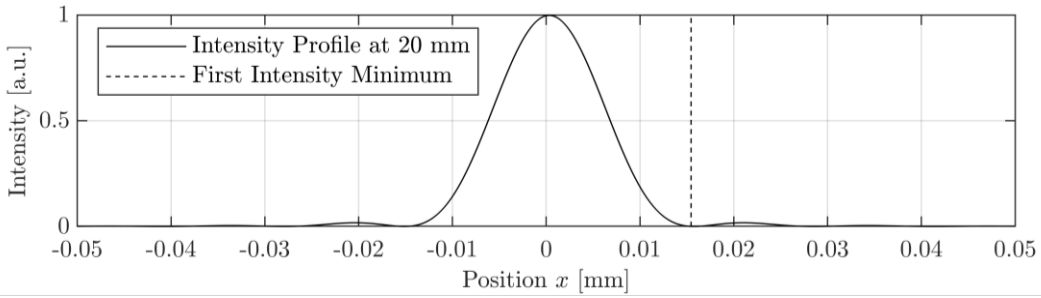


Figure 5-11: Intensity profile of a simulated focal spot in the focal plane. The aperture size is $D = 1\,mm$, the focal length of the lens is $f' = 20\,mm$, the wavelength is $\lambda = 633\,nm$ and the spatial sampling is $\Delta s = \lambda/2$.The position of the first intensity minimum to the right of the cnter is indicated by a dashed line.

The decentration can also be observed when looking at the data of Figure 5-10 in a logarithmic intensity scale. This is shown in Figure 5-12 with the white circle with a radius of $d_{airy} \cong 15.4\,\mu m$ around the center. It can be observed that the deviation of the airy disk to the drawn circle is constant, this the spot is decentered.

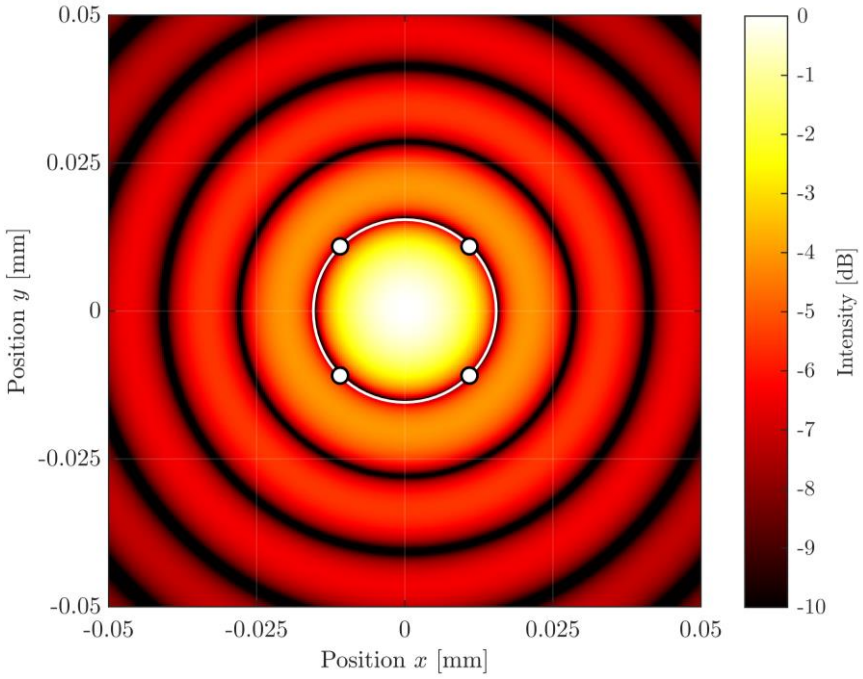


Figure 5-12: Intensity in the x-y-plane of a simulated focal spot in the focal plane. The intensity is shown in a logarithmic scale. The aperture size is $D = 1\,mm$, the focal length of the lens is $f' = 20\,mm$, the wavelength is $\lambda = 633\,nm$ and the spatail sampling is $\Delta s = \lambda/2$.

5.2.3 Discussion Part 1

The results of the the first part show that a optical field modulated by basic optical components can be simulated with the BLAS method. Four principles have been simulated. The focusing by lens yields predicted results. It is noteworthy that the lens modulation is calculated using a thin lens approximation, and the lens has to be sampled correctly. The later is not coverd by applying the sampling conditions of subsection 4.2.3. If the sampled lens function is subject to aliasing, outer parts of the aperture are refracted stronger than paraxial parts. This is basically pincushion type distortion, where image parts further away from the aperture have a stronger magnification than paraxial image parts. The focal point has been found at the correct position in z-direction. The decentration of the intensity is assumed to be an effect of choosing an even number of sampling point per field. The position vectors are shifted by the value of half a sampling distance to correctly show a zero. This shift, that is not considered is the output plot might be responsible for the decentration of the intensity images.

5.2.4 Experimental Setup Part 2

The second setup is the simulation of a diffraction grating. The first grating simulated is a binary amplitude grating with a slit width of exactly one sample distance $\Delta s = 2\,\lambda$ and a period of $d_{grating} = 100\,\mu m$. The expectations are multiple diffractive orders with a constant

amplitude due to the approximately infinite small slit width. The two first order diffraction maxima are expected to be at $x_{m=\pm1} = z \cdot \tan\left(\sin^{-1}\left(\pm\frac{1\lambda}{d_{grating}}\right)\right)$. The diffraction orders are focused by lens with a focal length of $f' = 60\,mm$, so that the observation plane is at $z = 60\,mm$. The setup is also shown in Figure 5-13. With the distance z the first order maxima are predicted to be at $x_{m=\pm1} \approx \pm 0.38\,mm$.

To verify the phase modulation, a binary phase grating is also simulated. The sampling distance Δs and the grating period $d_{grating}$ are the same as for the amplitude grating. The slit width or the duty cycle is half the grating period $d_{grating}/2$. For a maximum contrast, the phase modulation is 0 and π for low and high surface heights, respectively. The resulting diffraction pattern is expected to be without a 0^{th}-order and with an intensity drop for higher diffractive orders.

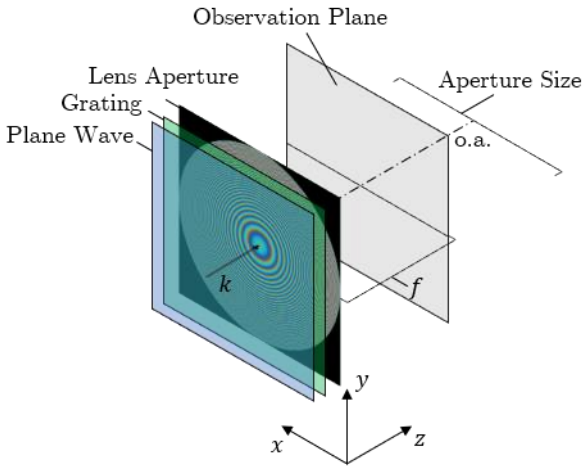


Figure 5-13: Setup for observing the diffractive pattern of a grating under illumination of a plane wave, that is additionally focused by a lens with the focal length f'.

5.2.5 Results Part 2

The results for the binary amplitude grating are shown in Figure 5-14 to Figure 5-16. The first image shows a cross-section in the x-z-plane around the focal point. The diffractive order maxima propagate as expected at an angle from optical axis.

To verify the diffraction angle, the intensity pattern in the focal plane is analyzed. Figure 5-15 shows the pattern and Figure 5-16 the profile at $y = 0$. The first order diffraction maxima $x_{m=\pm1} \approx \pm 0.38\,mm$ calculated by geometrical optics are indicated as dashed lines in Figure 5-16. They correspond with the simulation results.

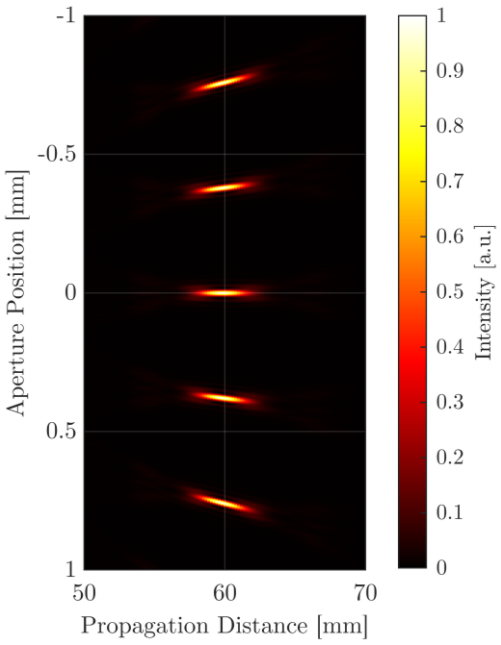


Figure 5-14: Intensity cross-section in the x-z-plane of the diffraction pattern caused by a binary amplitude grating and a focusing lens. Diffraction orders up to second orders are visible.

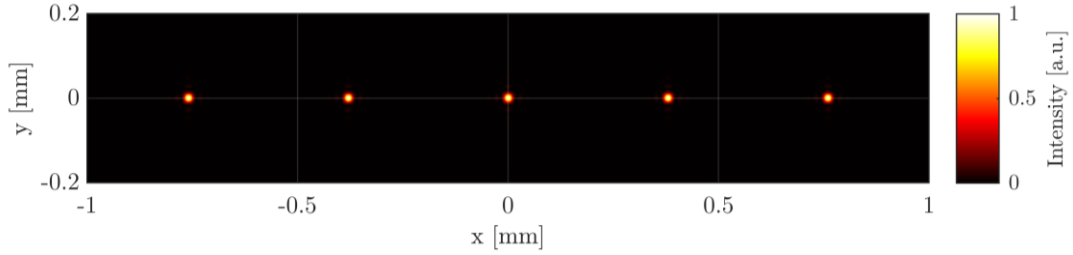


Figure 5-15: Intensity cross-section in the x-y-plane of the diffraction pattern caused by a binary amplitude grating and a focusing lens. Diffraction up to second orders are visible.

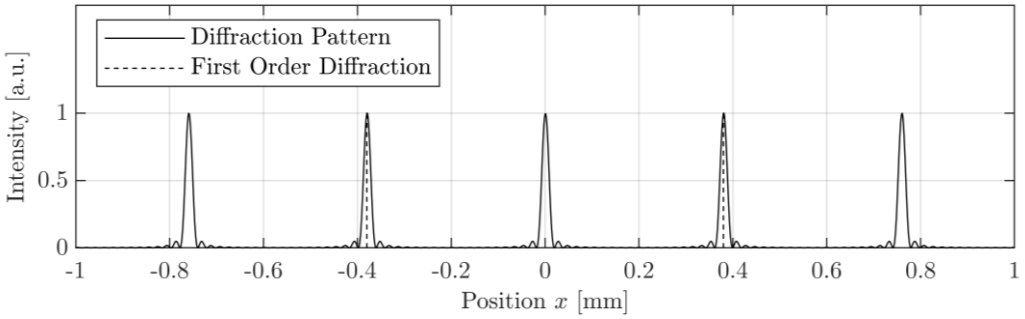


Figure 5-16: Intensity profile of the diffraction pattern caused by a binary amplitude grating and a focusing lens. Diffraction up to second orders are visible. The dashed lines indicate the calculated first order diffraction maxima.

The result of the phase grating simulation is shown in Figure 5-17. Only the $\pm 1^{th}$-orders are visbile, as expected. All other diffraction order are suppressed due to the physical expansion of the grating slits.

A cross-section at $y = 0$ of Figure 5-17 is shown in Figure 5-18 below. The expected first oder maxima are indicated by dashed lines. The logarithmic scale reveals the suppressed 0^{th}-order and $\pm 2^{nd}$-orders.

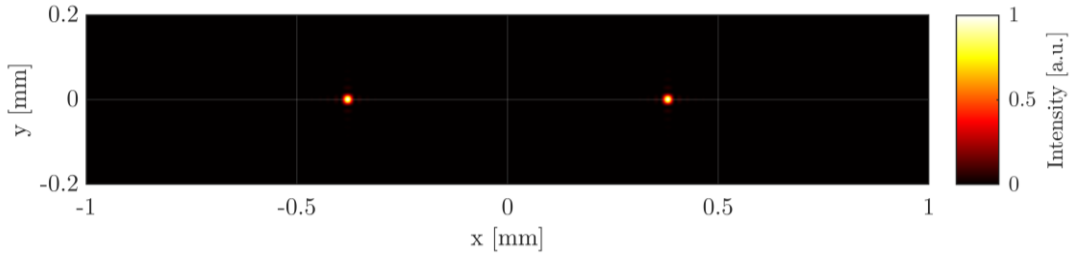


Figure 5-17: Intensity cross-section in the x-y-plane of the diffraction pattern caused by a binary phase grating and a focusing lens. Only the $\pm$ first orders are visible, the 0^{th}-order is suppressed.

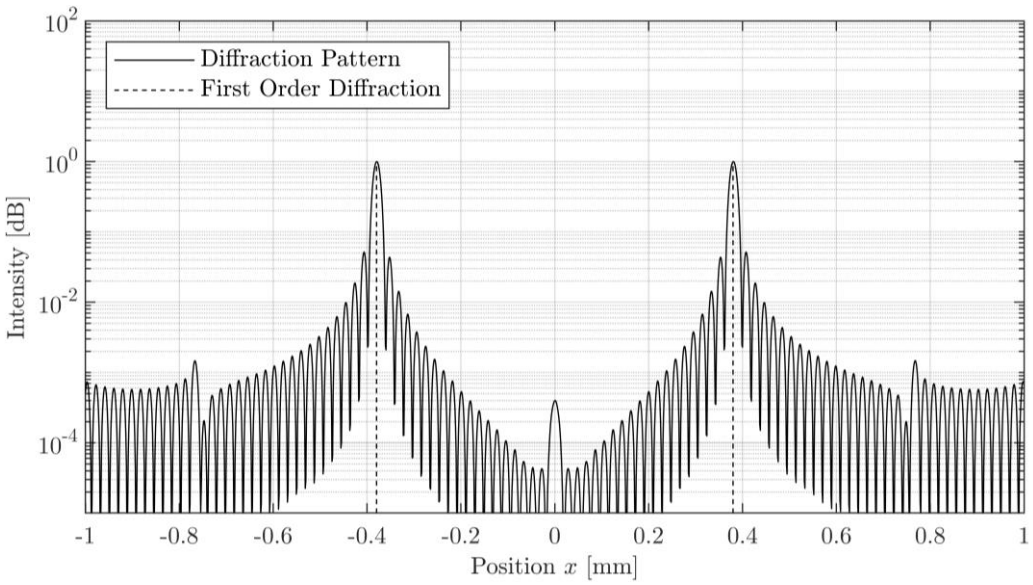


Figure 5-18: Intensity profile of the diffraction pattern caused by a binary phase amplitude grating and a focusing lens in a logarithmic scale. Diffraction up to second orders are visible. The dashed lines indicate the calculated first order diffraction maxima.

5.2.6 Discussion Part 2

The simulations of the binary amplitude phase diffraction gratings also yield the excepted results, as the diffracted orders are found at the predicted position. Furthermore, are effects of minimizing the slit width for binary amplitude gratings shown, whereby all diffracted orders have the same intensity. The 0^{th}-order suppression of an ideal phase grating is also shown in Figure 5-17 and Figure 5-18. Worth further investigations might a blazed grating, as it optimizes the diffraction efficiency.

5.2.7 Experimental Setup Part 3

In this experimental part, the basic convolution with a 4-f-setup is simulated. Therefore, the setup shown in Figure 5-19, composed of three planes is build. The input plane is focused by a lens with the focal length f' into the Fourier or focal plane. In the focal plane the spatial frequencies of the input image are spatially separated. A second lens, also with a focal length of f', collimates the beam again. Immediately after the second lens the resulting image is observed in the observation plane 1. Additionally, the image is viewed in the observation plane 2 which is at a distance of f' from the last aperture.

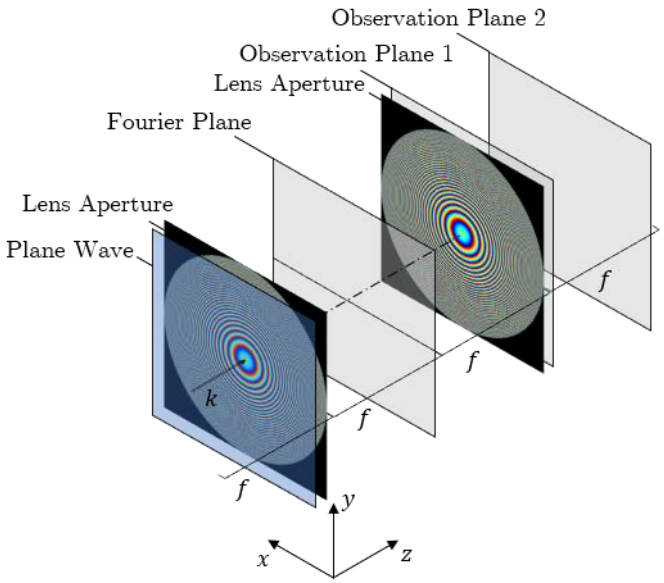


Figure 5-19: Experimental setup for a symmetric 4-f system with two lenses. The input is a plane monochromatic wave. The wave is first observed in the Fourier plane and directly behind the second lens aperture and the in the observation plane 2 at a distance f' form the last aperture.

The expectation for the output is that the same image appears inverted. Due to the use of monochromatic coherent light, transient oscillation at sharp edges also appears. In the Fourier plane a frequency image similar to a direct Fourier transformation should be observed. Further, in the Fourier plane the observed spectral bandwidth can be calculated using the diffraction angle of the highest spectral frequency that still fits the observation window. So, the highest spectral frequency in the Fourier plane can be derived using the grating equation [54, pp 56]:

$$d \cdot \sin(\theta_m) = \lambda \cdot m \tag{5-1}$$

Where d is the grating or structure period, m is the diffraction order and θ_m is the angle of which the incident light is diffracted.

Solving equation (5-1) for the reciprocal of d with respect to the first order diffraction $m = 1$, yields:

$$\frac{1}{d} = \frac{\sin(\theta)}{\lambda} = \frac{\sin\left(\tan^{-1}\left(\frac{x_{max}}{z}\right)\right)}{\lambda}$$

When the highest spatial frequency ν_{max} is the frequency which is on the outer edge of the spectral image in the Fourier plane, then $1/d = \nu_{max}$ and:

$$\nu_{max} = \frac{\sin\left(\tan^{-1}\left(\frac{x_{max}}{f'}\right)\right)}{\lambda} \tag{5-2}$$

Where x_{max} is the highest distance in the observation window in x-direction from the optical axis. If the computational window is square, then the maximum frequency in x- and y-direction is the same.

First, the sampling needed for a given computational window is evaluated. As in subsection 4.2.2, it is assumed that the simulated physical Fourier transformation with a lens also is subject to aliasing errors and therefore replicas of the spectral image in the Fourier plane. As shown in subsection 4.2.1, aliases do appear at a frequency $\nu_{rep} = \frac{1}{\Delta s}$ in frequency image due to the circular convolution. But when simulating a Fourier transformation by a lens, the highest frequency in the spectral image is not $\frac{1}{2\Delta s}$ as when using a DFT/FFT, rather it is defined by the geometrical relation of equation (5-2). Therefore, the sampling interval Δs must be chosen so that no replica falls into the Fourier plane image.

The computation window size in one direction is chosen to be $L_{x,y} = 20\,mm$. The focal lengths of the lenses are chosen to be $f'_{1,2} = 50\,mm$ the lens diameter $D = 20\,mm$, which results in a numerical aperture of $NA = \sin\left(\tan^{-1}\left(\frac{D}{2f'}\right)\right) \approx 0.196$. The highest frequency in x-direction in the Fourier plane is, according to equation (5-2). $\nu_{x,max} = \frac{\sin\left(\tan^{-1}\left(\frac{x_{max}}{f'}\right)\right)}{\lambda} = \frac{NA}{\lambda} \approx 309.82\frac{1}{mm}$ at a wavelength of $\lambda = 633\,nm$. Assuming that no optical power of aliases spills into the computation window if $\nu_{x,max} \leq \frac{1}{2\Delta s}$ the minimum sampling interval becomes $\Delta s \leq \frac{1}{2\nu_{x,max}} \approx 1.6\,\mu m$.

This sampling distance would result in a sampling point amount of $N = \frac{L_{x,y}}{\Delta s} = \frac{20\,mm}{1.6\,\mu m} = 12500$. The estimated matrix size for field becomes then with a padding size double the size of the

computation window: $(2 \cdot 12500)^2 \cdot 16\, bytes = 10\, Gb$ for double precision float complex numbers.

Nevertheless, for computational speed a lower spatial sampling interval with a large computational window size might be needed. A solution for this problem might be a limiting aperture in the Fourier plane. The aperture limits the contribution of higher frequency components, which the aliases are by nature.

In the first simulation, the aperture window is undersampled on purpose to examine the effects of aliasing, apodization and correct sampling. The sampling distances are:

- $\Delta s_1 = 10 \cdot \lambda = 6.33\, \mu m > \frac{1}{2\nu_{x,max}}$
- $\Delta s_2 = 5 \cdot \lambda = 3.165\, \mu m > \frac{1}{2\nu_{x,max}}$
- $\Delta s_3 = 2.5 \cdot \lambda = 1.5825\, \mu m < \frac{1}{2\nu_{x,max}}$

Where Δs_1 and Δs_2 are undersampled and Δs_3 is oversampled. The replicas resulting from the sampling distances Δs_1 and Δs_2 are predicted to be located at $x_{rep,1} = \tan\left(\sin^{-1}\left(\frac{\lambda}{\Delta s_{1,2}}\right)\right) \cdot f' \approx 5\, mm$ and $x_{rep,2} \approx 10.2\, mm$ from the optical axis, according to equation (5-2). The aperture truncating the spectral image in the Fourier plane must separate the spectral replicas. The point of separation is chosen to be in between the real spectrum and the aliases at the frequency $\frac{1}{2\Delta s}$, as shown in Figure 4-10 which corresponds to an aperture radius of $r_{AP,1} = \tan\left(\sin^{-1}\left(\frac{\lambda}{2\Delta s_{1,2}}\right)\right) \cdot f' \approx 2.5\, mm$ and $r_{AP,2} \approx 5\, mm$ for apodised cases. Although most of the spectral energy of the alias is then omitted by truncation, higher frequency components of the alias might still leak into the computation window. The results are shown in Figure 5-21, Figure 5-23, and Figure 5-24 for the at Δs_1 and Δs_2 undersampled and the at Δs_3 correctly sampled image, respectively and will be discussed in the next chapter.

The image that is being analyzed is the 1951 USAF resolution test chart with a magnification factor of $M_{USAF} = 0.2$. The USAF chart is also zero-padded so that the square image fits into the circular aperture of the lenses. The USAF chart is shown in Figure 5-20.

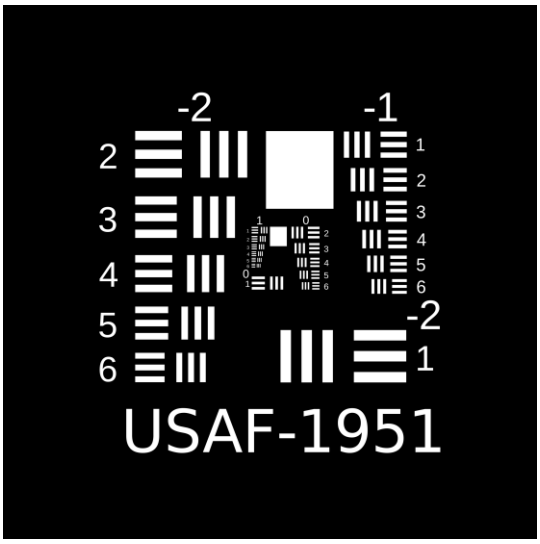


Figure 5-20: 1951 USAF resolution test chart used in this experiment with additional zero-padding.

5.2.8 Results Part 3

The simulation results for the 4-f system with the focal lengths of $5\,mm$ and a computational window size of $20 \times 20\,mm^2$ at a sampling distance of $\Delta s_1 = 10\,\lambda$ are shown in Figure 5-21. The intended undersampling of the aperture produces replicas in spectral and wraparound of image information in the spatial observation window. The predicted aliases at $x_{rep,1} \approx 5\,mm$ are clearly visible in Figure 5-21 a). Those replicas are, contrary to expectations, no exact copies of one another but seem to be unique. In Figure 5-21 b) the intensity image of the observation plane is shown. When comparing this image to the USAF chart of Figure 5-20, the assumption can be made that segments of the original image are dislocated and superimposed on one another.

In Figure 5-21 c) an aperture is applied in the Fourier plane with the radius of $r_{AP,1} = 2.5\,mm$. This radius is chosen to separate the central spectral image from the adjacent aliases. When truncated, the resulting intensity image in the observation plane only contains correctly located image portions, shown in Figure 5-21 d). But the image is missing parts at the outer edges. The valid area is squared different to the aperture which is circular.

In Figure 5-22 a) each dislocated image segment of Figure 5-21 b) is assigned a number and the corresponding segment in the inverted input image is shown in Figure 5-22 b). The computed image is expected to be inverted i.e., rotated by $180°$, but the alias order has the same orientation as the order in the object plane.

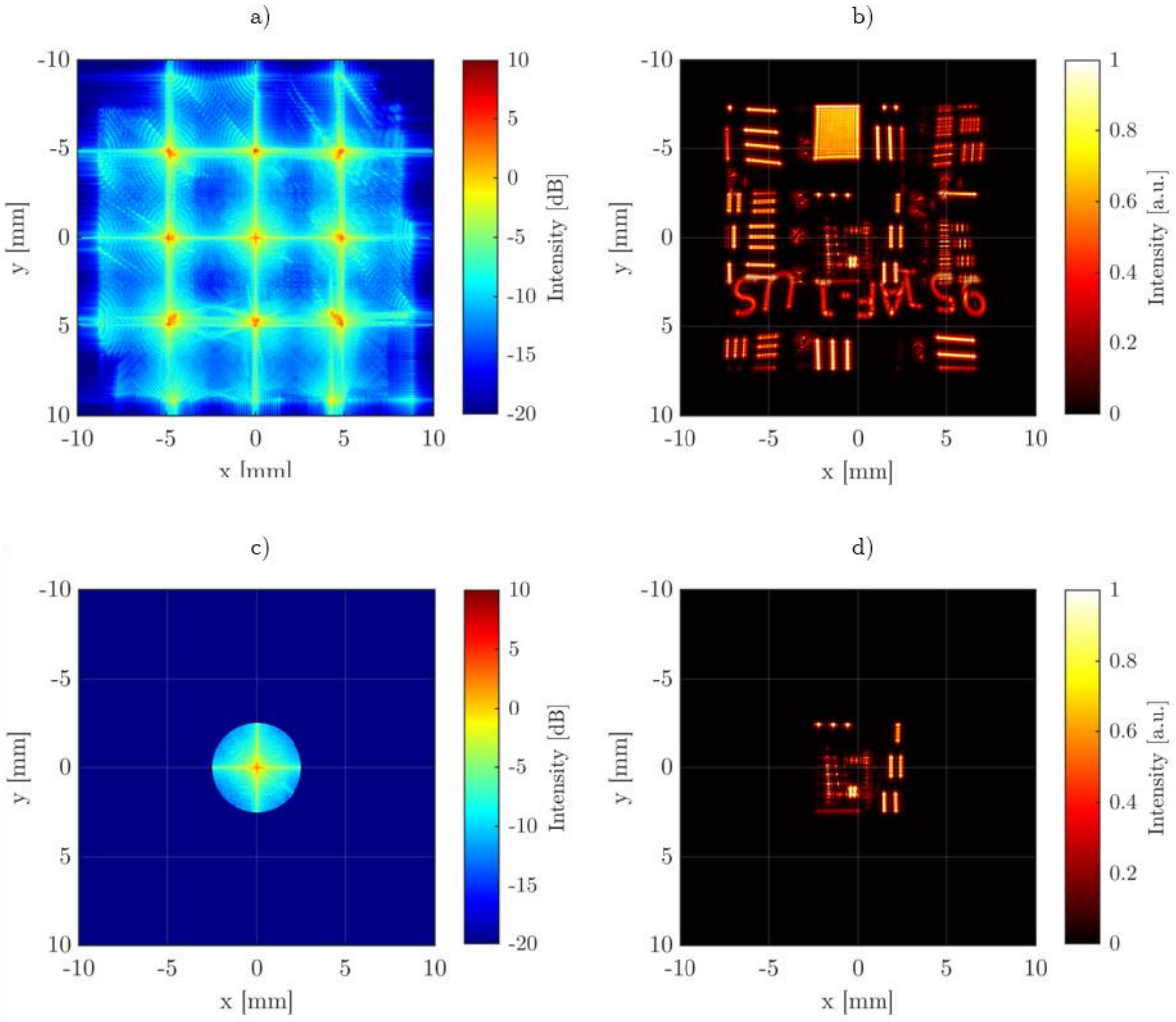


Figure 5-21: Simulation of a symmetric 4-f imaging system with the focal lengths $f' = 50\,mm$, a lens with the diameter of $20\,mm$ at a sampling distance of $\Delta s = 10\lambda$ and a simulation wavelength of $\lambda = 633\,nm$. In a) the intensity image in the Fourier plane is shown and in b) the corresponding intensity image at the observation plane is shown. In c) the spectral image is truncated by an aperture with the radius of $r_{AP,2} = 2.5\,mm$ and d) shows the corresponding image.

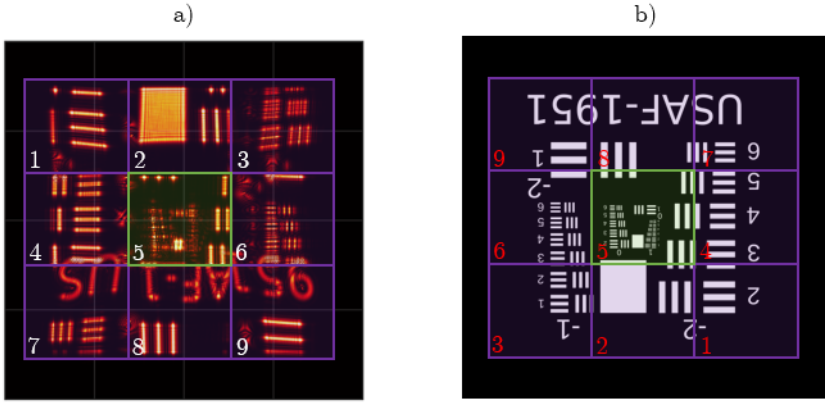


Figure 5-22: Fourier space alias locations. Whereby a) shows the simulation result of Figure 5-21 b) with numbered squares that roughly represent each image segment and b) shows the input image with the corresponding image segments correlated by numbers in both images.

With increasing resolution, the replicas in the frequency image are found at higher spectral frequencies as expected and shown in Figure 5-23 a). The expected position of the first replica in x-direction is at $x_{rep,2} \approx 10.2\,mm$. Qualitatively, the position corresponds to the simulated position of Figure 5-23 a). Although only the sampling distance $\Delta s = 5\lambda$ has been increased, the image in the observation plane shown in Figure 5-23 b) does image a larger portion of the USAF chart. Also, no superimposed dislocated image parts are visible. Further, there is a clear difference of the central spectral image visible.

If the spectral image in the Fourier plane is truncated with a circular aperture as shown in Figure 5-23 c) the intensity image in the observation plane Figure 5-23 d) shows no change to the not truncated image of Figure 5-23 b). An additional observation is that in Figure 5-23 b) and c) pincushion type image distortion is present.

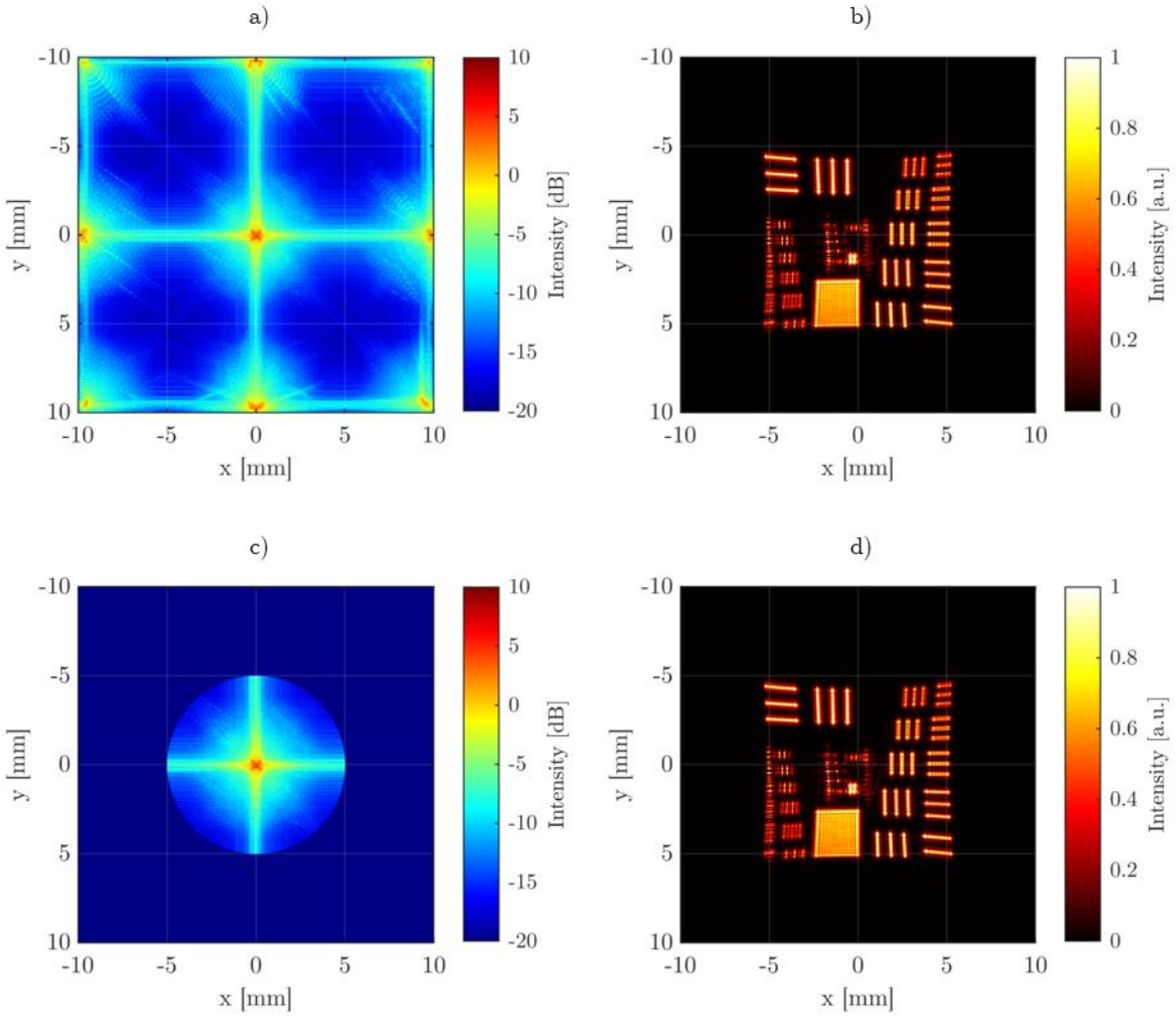


Figure 5-23: Simulation of a symmetric 4-f imaging system with the focal lengths $f' = 50\,mm$, a lens with the diameter of $20\,mm$ at a sampling distance of $\Delta s = 5\lambda$ and a simulation wavelength of $\lambda = 633\,nm$. In a) the intensity image in the Fourier plane is shown and in b) the corresponding intensity image at the observation plane is shown. In c) the spectral image is truncated by an aperture with the radius of $r_{AP,2} = 5\,mm$ and d) shows the corresponding image.

The results for an oversampled aperture are shown in Figure 5-24. The sampling distance hereby is $\Delta s = 2.5\lambda$ which is slightly smaller than the sampling condition from the discrete convolution theorem of $\Delta s_{crit} = \frac{1}{2\nu_{max}} \approx 2.53\lambda$. In the Fourier plane image of Figure 5-24 a) no aliases are visible. The intensity image in the observation plane in Figure 5-24 b) also shows no wraparound. Furthermore, is the complete USAF chart image visible, althought distortion is still present.

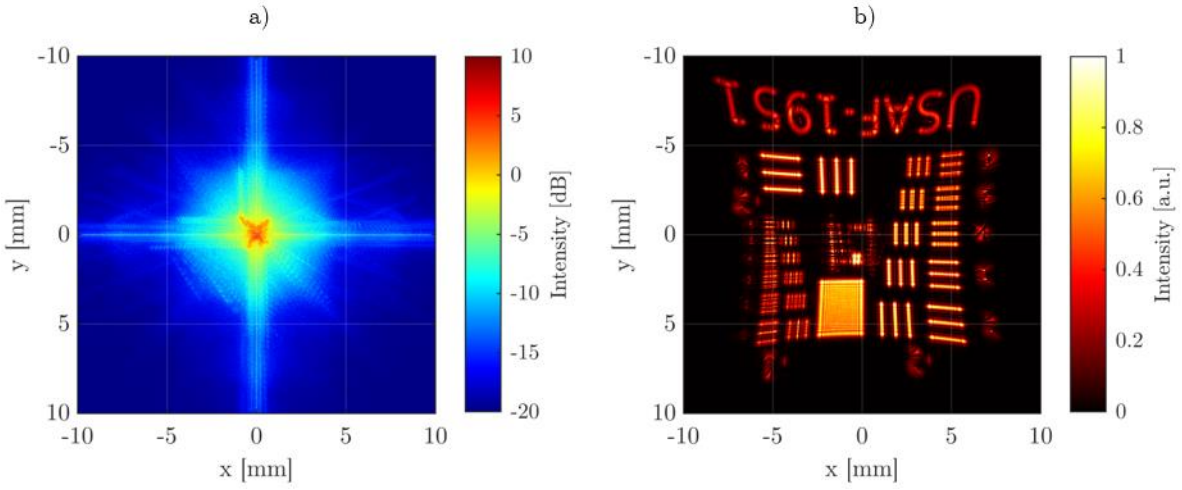


Figure 5-24: Simulation of a symmetric 4-f imaging system with the focal lengths $f' = 50\,mm$, a lens with a diameter of $20\,mm$ at sampling distance of $\Delta s = 2.5\lambda$ and a simulation wavelength of $\lambda = 633\,nm$. In a) the intensity image in the Fourier plane is shown and in b) the corresponding intensity image at the observation plane is shown.

Next, the Fourier plane images for different focal lengths are compared. The results are shown in Figure 5-25. The focal lengths simulated are $f_1' = 25\,mm$ in Figure 5-25 a) and b), $f_2' = 50\,mm$ in Figure 5-25 c) and d), and $f_3' = 100\,mm$ in Figure 5-25 e) and f). The left column of Figure 5-25 shows the FFT of the USAF chart in Figure 5-20, whereas the right column shows the simulated focal planes with the corresponding lenses. The FFT images are scaled with equation (5-2) to match the Fourier images. The spatial sampling of all images is $\Delta s = 5\lambda$ with a lens diameter of $D = 20\,mm$ and a simulation wavelength of $\lambda = 633\,nm$.

The Fourier plane in Figure 5-25 b) for the focal length of f_1' shows obvious effects of undersampling, because multiple spectra with the distance of $\frac{1}{\Delta s}$ appear. But due to the high numerical aperture of $NA = \sin\left(\tan^{-1}\left(\frac{D}{2 \cdot f'}\right)\right) \approx 0.37$ and therefore high field curvature and distortion is present in the focal plane. Due to this distortion the outer aliases are out of focus and not imaged correctly. But this effect allows one to observe that the images of the outer aliases are in fact images of the outer segments of the USAF chart and not just multiple images. For the focal length of f_2' the image in the Fourier plane shows still aliasing but with much

lower distortion, shown in Figure 5-25 d). Also, the higher frequency components are imaged more correctly in respect to the smaller focal length f_1', in comparison to the FFT image of Figure 5-25 c). The lower frequency components in the center still differ from the corresponding FFT image. By increasing the focal length to $f_3' = 100\,mm$ the spectral image in Figure 5-25 f) seems to be imaged correctly in comparison to the FFT image in Figure 5-25 e).

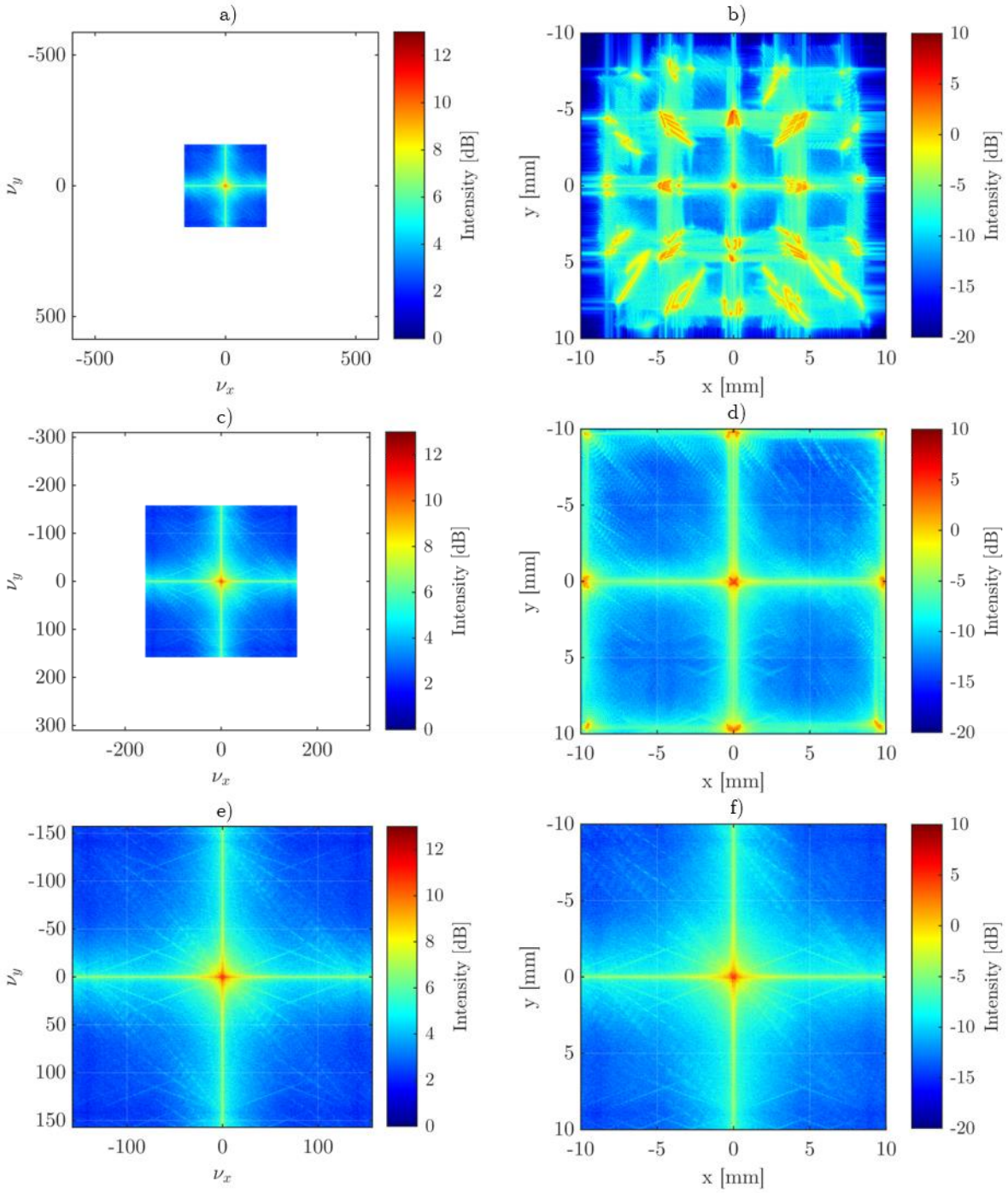


Figure 5-25: Intensity images of Fourier transformations of the USAF chart using the FFT in a), c), d) and a simulated lens and free-space propagation in b), d), f). The FFT images are scaled to show the same frequency band as the simulated images and are shown in units of $[mm^{-1}]$. *The focal length of the lens in a) is* $f_1' = 25\,mm$, *in c) the focal length is* $f_2' = 50\,mm$, *in e) the focal length is* $f_3' = 100\,mm$.

To investigate the distortion aberration that occurred in Figure 5-21 to Figure 5-24, the image is additionally computed $100\,mm$ behind the second lens of the system, which corresponds to the observation plane 2 in Figure 5-19. The simulation paramters are again $\Delta s = 2.5\,\lambda$, with a window size of $20\,mm$ and focal lengths of $f' = 100\,mm$. The resulting image is shown in Figure 5-26. The pincushion distortion has vanished, instead a slight barrel distiortion is present.

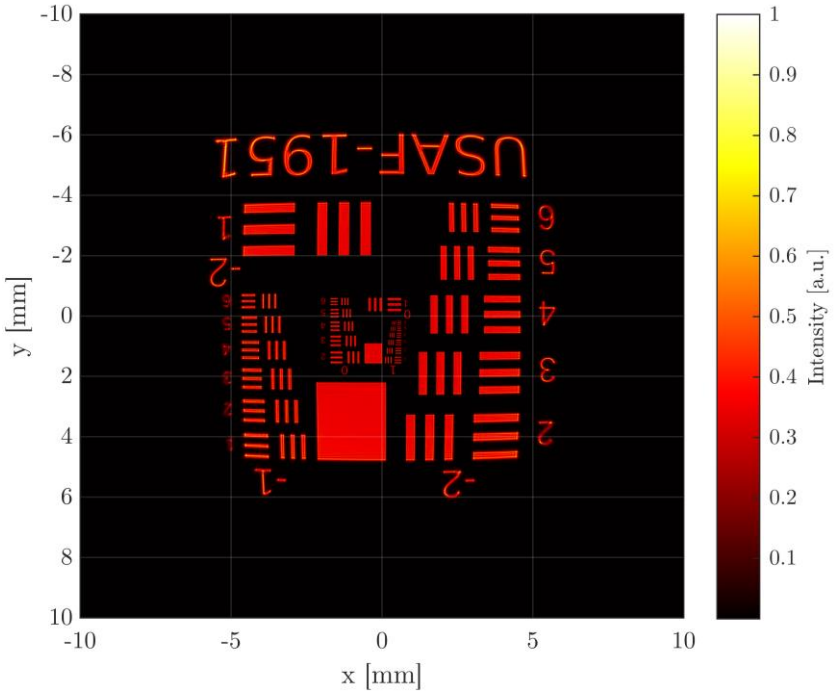


Figure 5-26: Simulated image of a USAF chart 100 mm behind a 4-f system with the focal lengths of $f' = 100\,mm$. The sampling is $\Delta s = 2.5\,\lambda$ and a window size of $20 \times 20\,mm^2$.

5.2.9 Discussion Part 3

The results of the previous subsection 5.2.8 show that there is a valid region for simulation depending on the spatial resolution chosen, which is assumed to be result of a undersampled lens aperture. The principle is shown in Figure 5-27 whereby a) shows the quarter section of a lens aperture which has a small sampling distance of $\Delta s = 3.2\,\mu m$. Only small aliasing shadows are observable at the edges. In b) the sampling distance is $\Delta s = 63.4\,\mu m$, which is much greater so that the lens aperture is undersampled. Due to the undersampling, lens-like aliases appear in outer parts of the aperture. These aliases cause different parts of the object plane to be imaged in separate locations in the image plane. To find a sampling condition that prevents undersampling of lens apertures, the sampling distance and the correctly imaged window size have to be related by a further sampling condition.

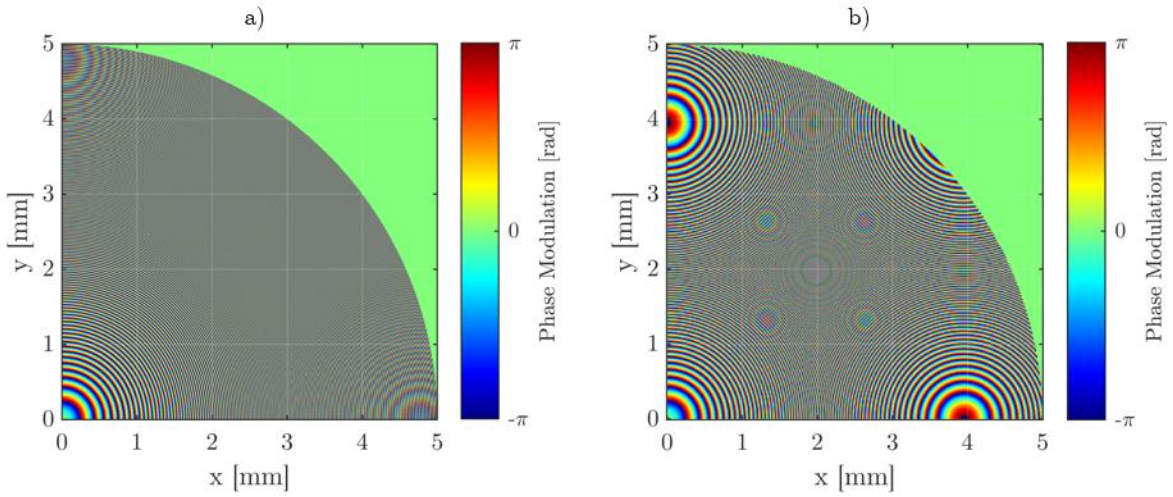


Figure 5-27: Quarter sections of lens apertures with a focal length of $f' = 100\,mm$. In a) a lens is shown which is sampled at $\Delta s = 3.2\,\mu m$ and in b) the sampling distance is $\Delta s = 63.4\,\mu m$.

With smaller sampling distance Δs the observable portion of the image increases. It can be concluded that the sampling interval limits the observable window size. These two are connected in the way that the spectral sampling distance is the reciprocal computational window size $L_{x,y} = \frac{1}{\Delta \nu_{x,y}}$ in the analytical description of the discrete Fourier transformation. The spectral sampling $\Delta \nu$ depends on the spatial sampling Δs by:

$$\Delta \nu = \frac{\nu_{max}}{\frac{L}{2\Delta s}} = 2\frac{\nu_{max}\Delta s}{L} \tag{5-3}$$

Where the maximum frequency in the Fourier plane[34] ν_{max} is calculated by equation (5-2). Inserting $\Delta \nu = \frac{1}{L}$ and equation $NA = \sin\left(\tan^{-1}\left(\frac{x_{max}}{f'}\right)\right)$ into (5-3) yields:

$$NA \leq \frac{\lambda}{2\Delta s} \tag{5-4}$$

Where NA is hereby the maximum numerical aperture observable at a given sampling distance for imaging simulations. Surprisingly, equation (5-4) is an inverse Abbe diffraction limit [77]. When giving this outcome a second thought, the backwards definition of an allowable numerical aperture depending on the image sampling makes sense as the sampling structure should not be resolved in the calculated image. However, this limitation only applies to the simulation of imaging optics, for the case of holographic diffractive optics without lenses and collimated illumination the Abbe diffraction limit might not be applied as $NA \to 0 \Rightarrow \Delta s \to \infty$.

[34] The highest frequency ν_{max} here is not to be mistaken with the bandwidth limit of the BLAS method of subsection 4.2.3.

Next, the results of the varying focal length in Figure 5-25, show that when the thin lens approximation is violated, then the focal length is not constant over the computational window. The thin lens approximation, which was used to calculate the focal lengths is [78]:

$$\frac{1}{f'} = (n_2 - n_1)\left(\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n_2 - n_1)d}{n_2 R_1 R_2}\right) \cong (n_2 - n_1)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \; \forall \; d \ll R_1, R_2 \qquad (5\text{-}5)$$

Where n_1 is the refractive index of the lens medium, n_2 is the refractive index of the surrounding medium, R_1 and R_2 are the radii of the lens surfaces and d is the lens thickness. For short focal lengths with the given lens diameter this approximation does not hold. The stronger curvatures of R_1 and R_2 cause a difference in the actual OPL through the lens. The conclusion is that for simulating real lenses with short focal lengths the approximation of equation (5-5) can not be made. But when simulating DOEs, as in this case a Fresnel lens, the simulation results are correct and the approximation of (5-5) images the real effects of the design approximation of a Fresnel lens. For correct simulation one must distinguish between real lenses and diffractive lenses and choose the correct calculation to approximate the OPD through the respective optical element.

A second observation of the results from Figure 5-25 is that when the thin lens approximation is obeyed the simulated image in the Fourier plane is similar to the FFT image. Furthermore, corresponds a change in focal length to a scaled FFT image. The scale factor can be derived using equation (5-2) and the frequency scale of the FFT from subsection 4.2.1. By relating the maximum frequencies of the image calculated using the FFT and of the simulated image one obtains:

$$M_{FFT} = \frac{\nu_{max,FFT}}{\nu_{max,Sim}} = \frac{\lambda}{2\Delta s \cdot \sin\left(\tan^{-1}\left(\frac{D}{2f'}\right)\right)} = \frac{\lambda}{2\Delta s \cdot NA} \qquad (5\text{-}6)$$

By using this scale factor one can hypothesize that an imaging lens can be simulated by using the BLAS method or by using a scaled FFT of the input image, if the conditions of equation (5-3) and (5-4) are met. The results of Figure 5-25 show that this is true if the scale factor $M_{FFT} \geq 1$. Therefore, restating equation (5-6) as:

$$M_{FFT} = \frac{\lambda}{2\Delta s \cdot NA} \; \forall \; M_{FFT} \geq 1 \qquad (5\text{-}7)$$

Using this scaled FFT approach is equivalent to using the Fresnel approximation for the propagation between two spherical planes [32, pp 66–74], which holds true for paraxial approximations and structure size larger than the operation wavelength. Applying and testing

the Fresnel approximation for convolutional units is out of the scope for this thesis but may be further investigated to speed up the calculation of propagation between two spherical lenses. Nevertheless, the equation does give an additional sampling condition for Δs, so that no aliases appear in the focal plane due to simulation sampling:

$$\boxed{\Delta s \leq \frac{\lambda}{2NA}} \tag{5-8}$$

Which again is the inverse Abbe resolution limit. Relating this result back to Hypothesis 3 of this thesis, one might suggest that a micro lens array can be subdivided into one micro lens and if the multiple images created by the previous grating are exact copies, then the multiple kernel calculation may also be performed using separate BLAS calculations instead of computing the complete field with the BLAS method. The complete image may than be stitched together afterwards.

An additional observation of this experimental part is, that the near-field image differs from the far-field image. Figure 5-26 shows that the oscillations at sharp edges are smaller in amplitude and have a higher frequency. The image uniformity and resolution have increased in contrast to the images exactly behind the lenses in Figure 5-21 to Figure 5-24. This has to be further investigated.

5.2.10 Experimental Setup Part 4

To show that a multi-kernel diffractive unit can be realized theoretically, the principle is examined by using a BLAS simulation. The setup follows Figure 5-28, in which a collimated image hits a transmission grating, and the first diffraction orders are focused by a lens. In the focal plane each wave path might be modulated by a separate kernel and a lens array collimates the divergent beams. In the image plane, a subsequent convolution unit or a feed-forward network might be placed. In this third experimental part, the imaging by a lens as well as the diffraction by a grating from the first part and the Fourier plane imaging with the condition applied by equation (5-8) are brought together. The system as a whole will be tested, and the similarity of the multiple diffractive orders examined. If all images are exact multiple copies of each other the computation might be split into parts to reduce the computational window size necessary. First the system as a whole is examined with the parameters as follows.

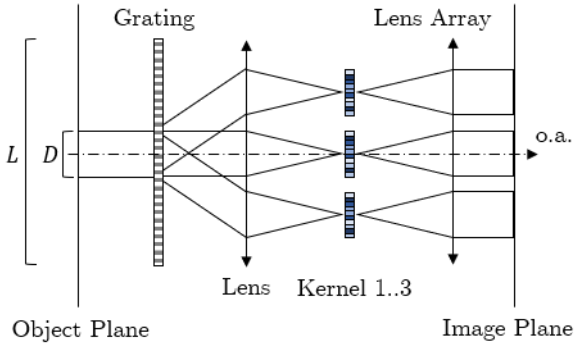


Figure 5-28: Conceptual design of a multi-kernel convolutional unit for a D^2NN.

The computational window is chosen to be the size of $L = 10\,mm$. The input is a plane wave truncated by a circular aperture with the diameter of $D = 2\,mm$ shown in Figure 5-29 a). An initial base zero-padding of factor two, provides a sufficient spectral resolution:

$$L_P = 2 \cdot L = \frac{1}{\Delta v} = 20\,mm$$

The goal focal length for the systems components is chosen to be $f' = 100\,mm$, which is then the propagation distance Δz from grating to lens, lens to Fourier plane, and so on.

The bandwidth limit ν_{max} for the BLAS window becomes:

$$\begin{aligned}\nu_{\max} &\leq \frac{1}{\lambda\sqrt{4\Delta\nu^2\Delta z^2 + 1}} \\ &= \frac{1}{633\,nm\sqrt{4\left(\frac{1}{20\,mm}\right)^2 (100\,mm)^2 + 1}} \\ &= 157.19\,\frac{1}{mm}\end{aligned}$$

Using the rule of subsection 4.2.3 for the spatial sampling:

$$\Delta s \leq \frac{1}{2\nu_{max}} = \frac{1}{2 \cdot 157.19} mm = 3.18\,\mu m$$

The Abbe resolution limit of equation (5-8) gives a spatial resolution limit of:

$$\Delta s \leq \frac{\lambda}{2NA} = \frac{\lambda}{2 \cdot \sin\left(tan^{-1}\left(\frac{L}{2f'}\right)\right)} = \frac{633\,nm}{2 \cdot \sin\left(tan^{-1}\left(\frac{20\,mm}{2 \cdot 100\,mm}\right)\right)} = 3.18\,\mu m$$

Comparing the two results above yields that the sampling conditions are identical. Thus, in case of a lens as imaging aperture function, following relation is true:

$$\frac{1}{\sqrt{4\Delta\nu^2\Delta z^2 + 1}} = NA \tag{5-9}$$

Using this spatial resolution automatically limits the resolution of the diffraction grating. In this experiment a binary phase grating is modelled. Aiming for a first order diffraction angle of $\varphi_{\pm 1} = \tan^{-1}\left(\frac{2.5\ mm}{f'}\right)$, so that the diffracted images are spaced $2.5\ mm$ apart with a $0.5\ mm$ gap between each image, yields a grating period of $d = \lambda / \sin(\varphi_{\pm 1}) \approx 40\lambda$ from equation (3-9). When sampling a grating with 8 points per grating period, one should not expect a high grating efficiency, as the results of this experiment will show. This means that the 0^{th} diffraction order will still be dominantly present and higher orders have less intensity as they would be using a real binary phase grating. Yet, using an amplitude grating should yield similar results to this setup. In reality, a highly ideal phase grating with a smaller diffraction angle could be used as the center of the output plane would be occupied by four diffraction orders in contrary to one single order. However, this is changing the kernel density per area, but it gives means to choose an even or odd number of kernels. The fact that diffraction efficiency differs from reality is therefore ignored in all further context.

The first simulation is a calculation of the complete three-dimensional optical field between the modulating layers of the grating, the lens, and the lens array. In propagation direction the resolution is $\Delta z = 1.01\ mm$. The modulating planes are shown in Figure 5-29 a) to c), whereby a) is the circular input, b) is a magnified image of the binary phase grating, c) is the collimating lens with a focal length of $f' = 100\ mm$ and d) is the lens array in which the estimated positions of the diffracted orders are indicated by the crossing of two dashed lines.

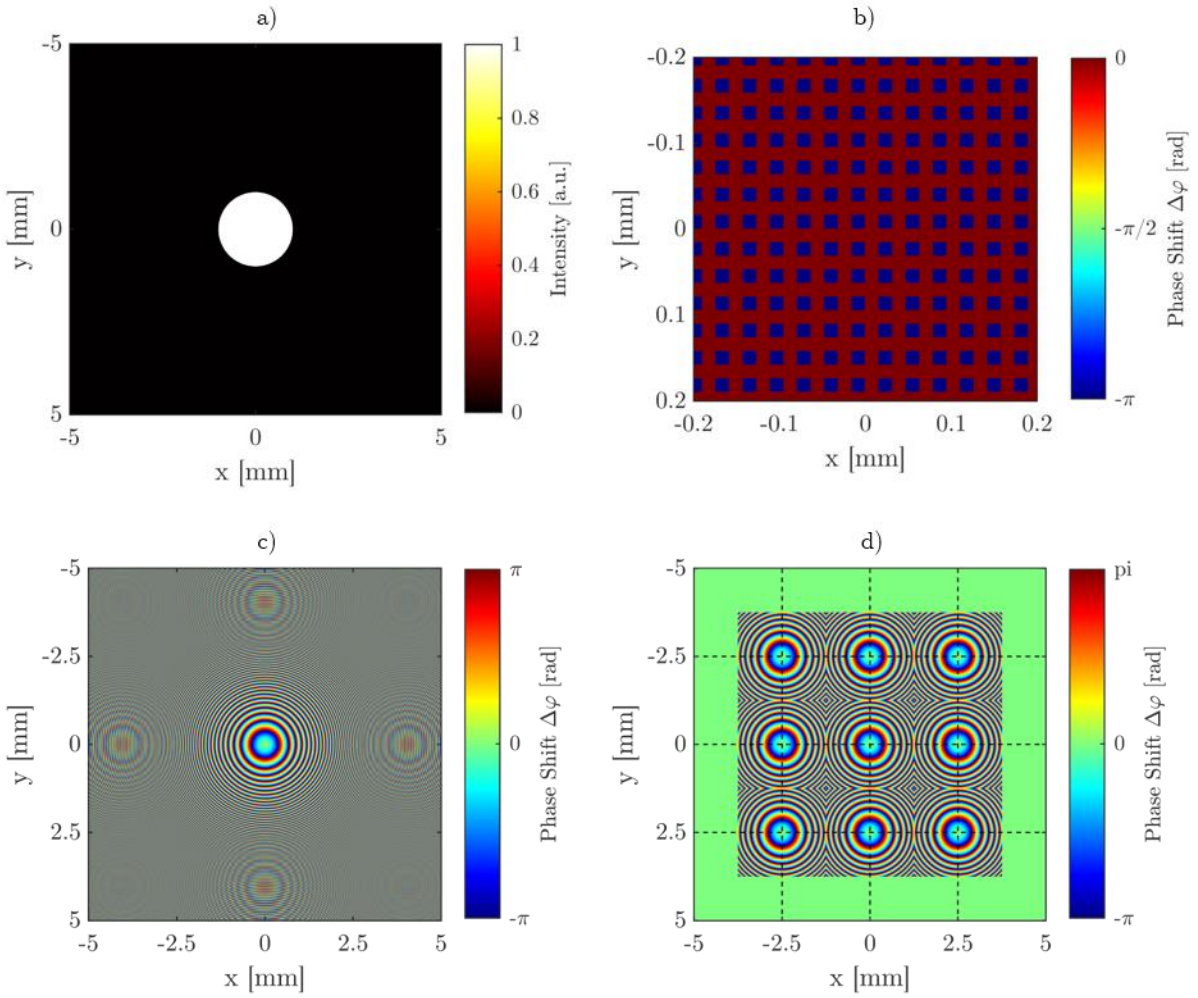


Figure 5-29: Images of the modulating planes in the third experimental setup. The circular aperture function with the diameter of $2\ mm$ *is shown in a). A magnified image of the binary phase grating is shown in b). The phase of the focusing with* $f' = 100\ mm$ *lens is shown in c). The lens array with a lens displacement of* $2.5\ mm$ *is shown in d).*

The second simulation is a comparison of the multiple images in the image plane of Figure 5-28. Therefore a down scaled version of the USAF chart in Figure 5-20 is used as input image. The down scaled version is also depicted in Figure 5-30 and is scaled so that image replicas do not overlap in the image plane. The parameters for the simulation are $\Delta s = 5\lambda$ and $\lambda = 633\ nm$. The computed window has the size of $20\ mm$, the focal length of the lens and each lens of the lens array is $f_1' = 100\ mm$ and the grating period is $\frac{1}{4 \cdot ds} = \frac{1}{12.66} mm^{-1}$. The grating pattern is the same as in Figure 5-29 b). The resulting distance of image multiples is $5\ mm$. The intensity and phase in the image plane are qualitativly compared.

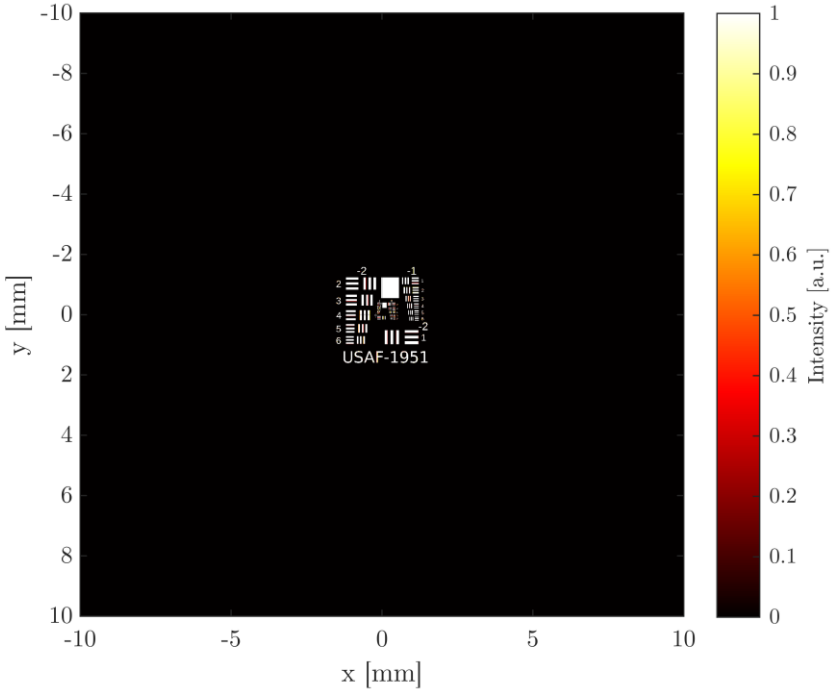


Figure 5-30: Scaled USAF chart the testing the simulation of a multi-kernel convolution unit.

5.2.11 Results Part 4

The results of the three-dimensional simulation are shown in Figure 5-31 and Figure 5-32, as a cross section at $y = 0$. In Figure 5-31 it is shown that the grating diffracts the incoming wave as expected and the multiple images in the lens' focal plane have a distance of $2.5\,mm$ in x-direction. The lens focuses the multiple images in a common focal plane and the lens array at two focal lengths from the lens then collimates each image.

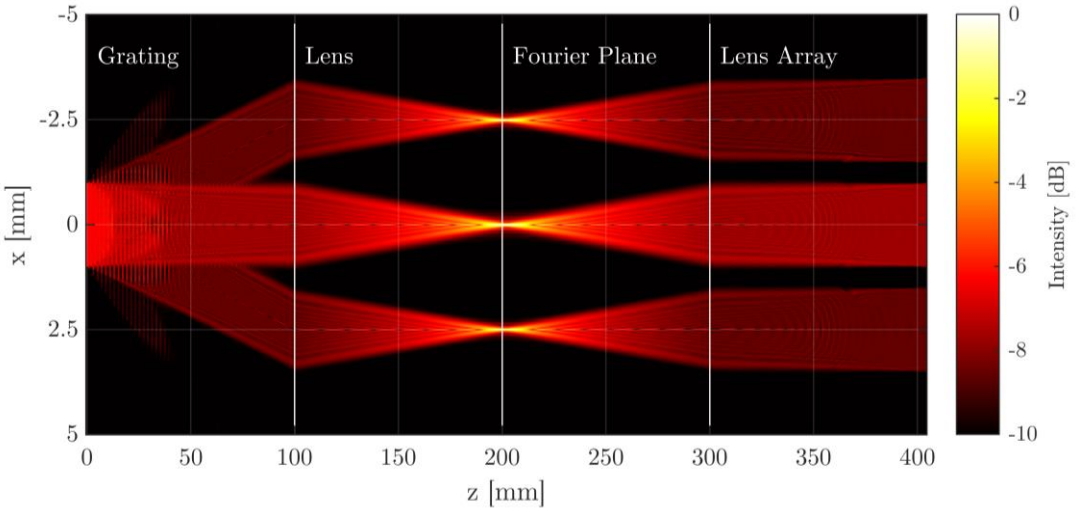


Figure 5-31: Cross section of a three dimensional BLAS simulation of a conceptual convolution unit. The input is a plane wave truncated by a $2\,mm$ aperture which is modulated by a binary phase grating. A lens at a distance of $100\,mm$ and with a focal length of $f' = 100\,mm$ focuses each image in the Fourier plane. A lens array located $2 \cdot f'$ from the aperture collimates each image separatly.

The Fourier image at the focal plane is viewed more closely in Figure 5-32. The dashed line indicates the plane of maximum intenstiy of the 0^{th} order is located, which is at $z = 202\ mm$. The calculated focal plane with respect to the propagation distance sampling is at $z = 200.02\ mm$. This corresponds to a difference of approximatly $2\ mm$.

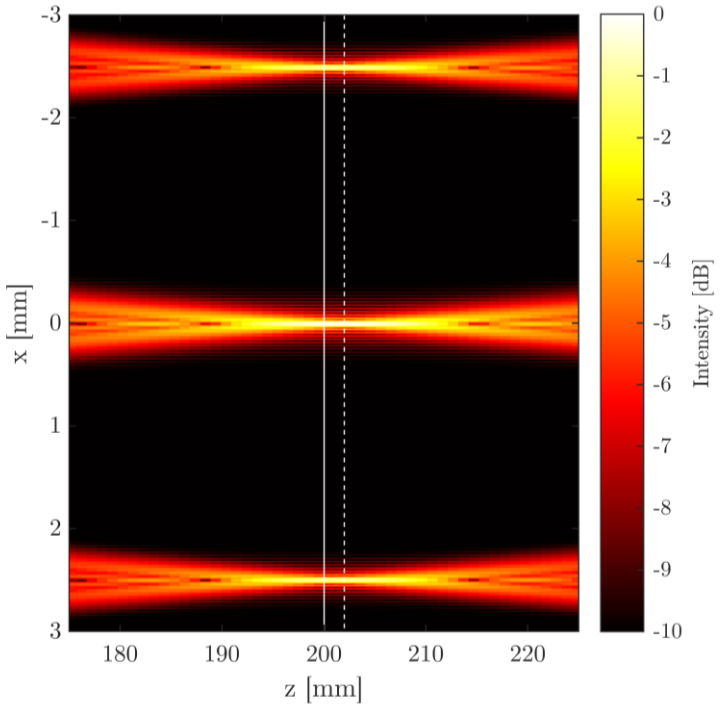


Figure 5-32: Magnified image of the Fourier plane in Figure 5-31. The theoretical focal plane is indicated as solid white line at $z = 200.02\ mm$ and the measured focal as dashed line at $z = 202\ mm$.

The results of the second simulation are shown in Figure 5-33 and Figure 5-34. In the intensity image of Figure 5-33 nine diffractive orders can be observed, each with a distance of $5\ mm$ to the 0^{th} order diffraction. Due to the undersampling of the phase grating with four samples per period, the 0^{th} order is visible and higher orders are weaker in intensity. Besides the obvious intensity difference, no deviation might be pointed out.

The phase of the optical field in the image plane is shown in Figure 5-34. In contrary to the intensity image of Figure 5-33, the phase map reveals that the wavefront is asymmetric. Each multiple image is no exact copy of the 0^{th} order image, as the overall offset is radial symmetric but the images are shifted and not mirrored.

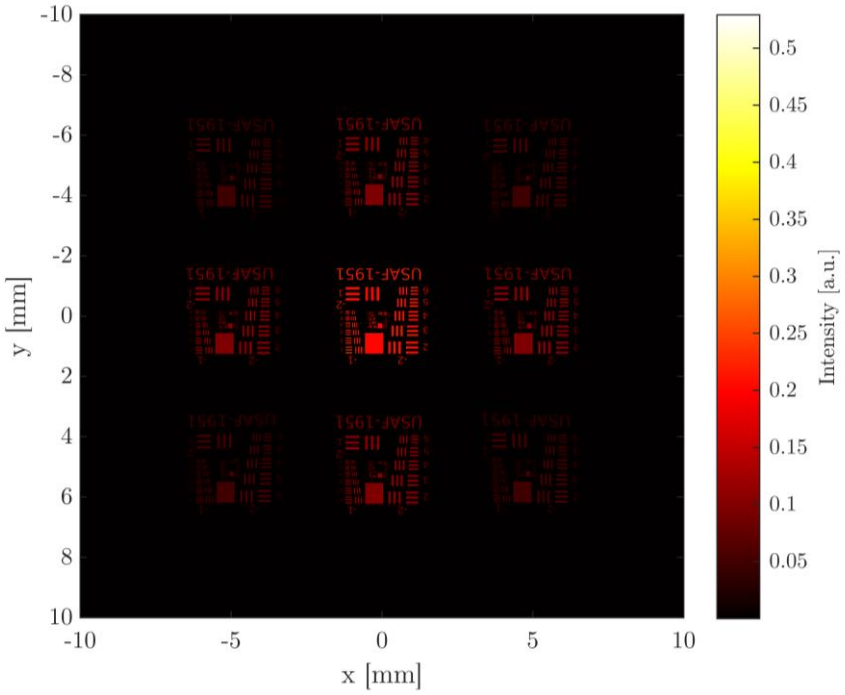


Figure 5-33: Intensity image of a USAF imaged by a convolutional unit, with the focal lengths of $f' = 100\,mm$, a spatial sampling of $\Delta s = 5\lambda$, a window size of $10 \times 10\,mm^2$ and a binary phase diffraction grating.

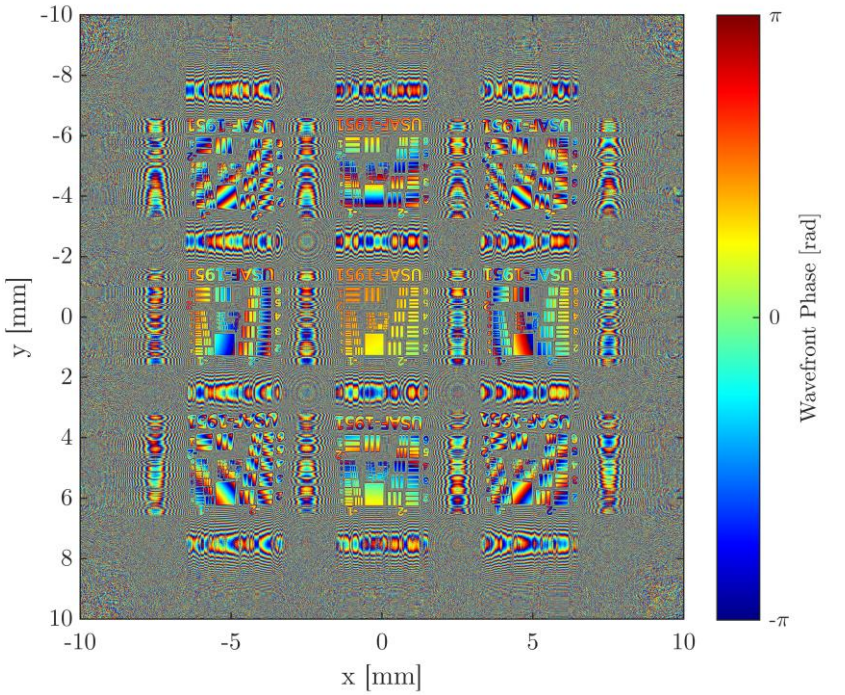


Figure 5-34: Phase image of a USAF imaged by a convolutional unit, with the focal lengths of $f' = 100\,mm$, a spatial sampling of $\Delta s = 5\lambda$, a window size of $10 \times 10\,mm^2$ and a binary phase diffraction grating.

5.2.12 Discussion Part 4

The results of this third experimental part show that indeed an optical system build of a diffraction grating, a traditional lens and a lens array system can produce multiple frequency

images in the Fourier plane. This is the basis for a convolutional D^2NN, in which spatial variant objects are transformed into a single intensity spot at the output of several convolutional units. The pooling function described in section 2.4, might be realized by exploiting the diffraction limit. Further research on how the image can be condensed to a set of output spots must be made. The limitations of the system theoretically developed are the physical spacing between layers due to the diffraction angle of the grating and focal lengths of the lenses. The diffraction efficiency of the diffraction grating must be accounted for, because as lower power in outer diffraction orders cause an intensity offset in the respective output.

The second part of this third experiment shows that for the calculation of the forward propagation, each image multiple can not be calculated separately. The complete forward pass must be calculated as one image, as the images' phase is depended on the spatial location. The computation time for multiple kernels becomes large because the high frequency diffraction grating demands high spatial sampling, and the separation of image multiples cause a larger computation window size.

Although the calculation time increase, the derived method is only possible with reasonable effort using the BLAS method. Furthermore is this method distinct from the approach of [47], as it does not act as a image filter but rather it abstracts spatial feature of an image. Each feature is represented as one single value that might by fed into an optical or electrical feed forward neural network. This approach is mathematically closer to the classical CNN described in section 2.4.

5.3 Scalar Diffraction Simulation with Real Data

The analytical approach and discrete derivatation of the RS integral and AS method are based on the assumptions made by the FK integral. This purely theoretical scenario might make mathematical sense but lacks validation with real data. In this third experiment one scenario is simulated by the BLAS method described in chapter 4.2 and also measured in a physical experiment. The two approaches should be as equivalent in their environmental condition as possible. The scenario is a light field modulation by a surface, in this case a diffraction grating and propagation to a observation screen. The field is measured in front of the grating and fed as input into the simulation. Also, the diffraction grating is measured and used as modulation inside the simulation. The resulting interference pattern in the experiment and in the simulation are compared.

5.3.1 Setup

The setup is sketched in Figure 5-35. A monochromatic helium neon laser emitting at $\lambda = 632.8\,nm$ is chosen as illumination source. The laser beam is conditioned by a two lens collimator of f_1, f_2 and the aperture stop A_1. The collimated beam is truncated by a second aperture stop A_2. A beam splitter sends half of the incomming power on a Thorlabs shack-hartmann wave front sensor (WFS) [79]. The wave front sensor measures the local phase offset and intensity of a copy of the wavefront falling onto a chromium diffraction grating. The diffraction grating has grating lines with a pitch of $d_G = 7.5\,\mu m$. The wavefront gets modulated by the grating and propagates a distance $\Delta z = 30\,mm$ to a white diffusing screen on which the incomming light interferes. The interference pattern on the screen is observed by a camera.

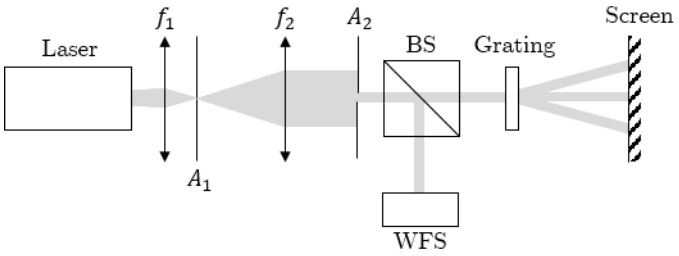


Figure 5-35: Experimental setup with a laser at $\lambda = 632.8\,nm$ is collimated and widened with two lenses f_1, f_2 and an aperture stop A_1. A second aperture stop A_2 truncates the beam for a homogenous intensity profil. With a beam splitter BS one path of the laser beam is measured with a wavefront sensor WFS and the other passes a grating and the interference pattern is observed at a diffusing screen.

The simulation reflects the light path from virtually[35] immediatly in front of the grating to the screen. One assumption made is that the incoming wave is absolute monochromatic. The second is, that grating has no real thickness. The grating pattern is etched on glass. The glass will produce at virtual axial position offset, which will be shifted to the screen. The thickness of the grating is $d_g = 1.51\,mm$. The third is, that the surrounding medium is absolutly homogenous and isotropic. The measured value is the diffraction angle of the diffraction orders and overal shape of the simulation compared to the measurement data. The propagation distance from grating to screen is $\Delta z = 30\,mm$.

The data given as input is the measured surface of the chromium grating, where grooves represent transparent parts, where the chromium is ecthed. The surface measurements are done by a white-light interferometer (WLI) from Zygo [80]. The grating topology is interpreted as

[35] Virtually, because the wavefront is measured at the WFS.

relative transparency value from 0 to 1, where transparency 0 corresponds to high surface values and vice versa. The other input data is the wavefront information measured by the WFS. The amplitude and phase is measured with 13 samples in x- and y-direction and with a sampling distance of $0.15\ mm$. The wavefront data is resized and interpolated by the MATLAB *imresize* function [73]. The interpolation method used is bicubic interpolation.

The simulation parameter setup is in accordance with the guideline of chapter 4.2.3. In addition the sampling of the aperture has to be considered. The needed screen size has to be estimated first. Therefore the equation of the diffraction angle for the negative first order maxima of a grating is calculated as:

$$\theta_{-1} = \sin^{-1}\left(\frac{\lambda}{d_G}\right) = 4.84° \tag{5-10}$$

With a screen distance of $\Delta z = 30\ mm$, the distance of the first order maxima would be $d_{-1,1} = 30\ \text{mm} \cdot \tan(4.84°) = 2.54\ mm$ theoretically. Because the laser beam is centered in the aperture window and the complete diffracted spot should be visible, a computation window size of $L_{x,y} = 2 \cdot 3.5\ mm$ is chosen. Padding the aperture to a minimum size of $P_{x,y} = L_{x,y} \rightarrow L_{P,x,y} = 14\ mm$, the maximal spatial frequency is according to equation (4-76) and (4-77):

$$\nu_{\max} = \frac{1}{\lambda\sqrt{4\Delta\nu^2 z^2 + 1}} = \frac{1}{632.8\ nm\sqrt{\frac{4 \cdot 30^2\ mm^2}{14^2\ mm^2} + 1}} = 359.086\frac{1}{mm}$$

From the summary of subsection 4.2.3 the biggest spatial sampling distance for accurate calculation can be determined to:

$$\Delta s \leq \frac{1}{2\nu_{max}} = 1.4\ \mu m$$

This would mean that the diffraction grating would be sampled with $\frac{7.5\ \mu m}{1.4\ \mu m} \cong 5.4$ points per grating period, which is hardly enough for a fourier series expansion to reconstruct a rectangular function. Further, for the sampling distance Δs causes a field size of $N_P^2 = \left(\frac{14\ mm}{1.4 \cdot 10^{-3}\ mm}\right)^2 = 10000^2$. Using double precision complex numbers the size of one matrix becomes:

$$N_P^2 \cdot 1.6 \cdot 10^{-8} \approx 1.6\ Gb$$

With available RAM of $49\ Gb$ the sampling distance Δs might be chosen smaller so that the structure of the diffraction grating is reflected better. So, sampling the structure at a sampling distance at $\Delta s = \frac{\lambda}{2}$ would yield a matrix size of $31\ Gb$. The RAM would only be able to hold

one operant at the same time. When optimizing for memory the approach is to reduce the double precision complex format to a single format prescision, which is half the size. The goal is a matrix size of approximately $10\ Gb$, so both operants and the result might be held in RAM.

$$N_P(10\ Gb) = \sqrt{\frac{2 \cdot 10Gb}{1.6 \cdot 10^{-8}}} = 35355$$

For fast FFT calculation a number close to 35355 and exactly 2^n is chosen, that would be $2^{15} = 32768$. The final sampling of the aperture is then:

$$\Delta s = \frac{14\ mm}{2^{15}} = 427.2\ nm$$

This sampling distance would sample the grating structure period with approximatly 18 points per period.

5.3.2 Results

Figure 5-36 a) shows the measured grating surface of $6 \times 6\ mm^2$. There are slight variations in the absolut height as relatively dark areas are visible. Figure 5-36 b) shows an enlarged section of a) with a defect at approximatly $x = 300\ \mu m$ and $y = 950\ \mu m$.

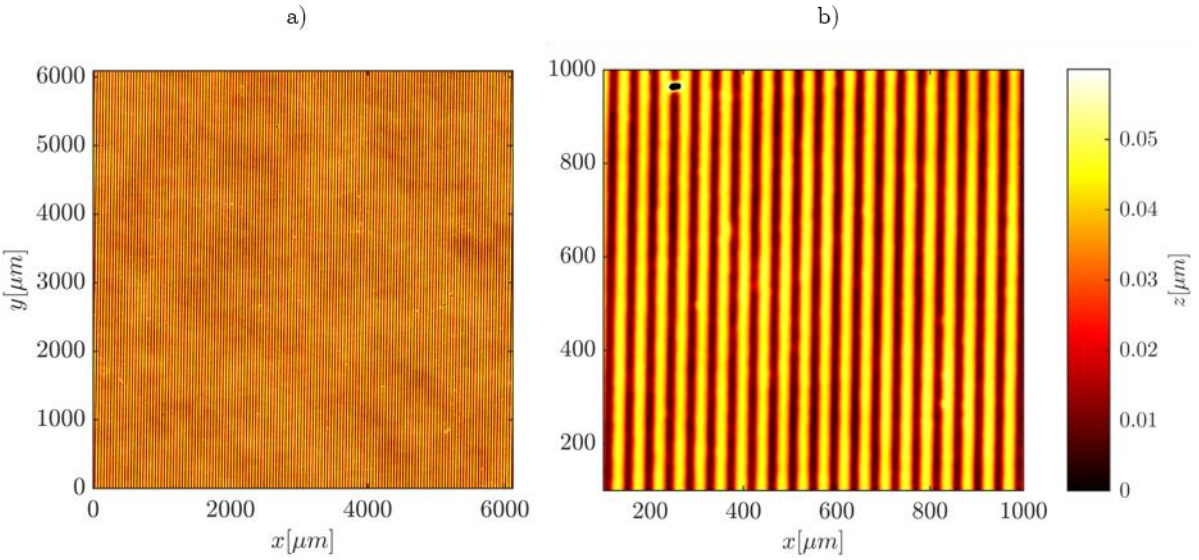


Figure 5-36: Surface topology of a chromium grating measured by a WLI. a) show the complet data of $6 \times 6\ mm^2$ and b) a subsection $0.9 \times 0.9\ mm^2$ of a) to visualize the individual grating lines.

The normalized topology of the grating is inverted to get a relative transmission value and cropped to reduce measurement artefacts at the edges. The whole data set is shown in Figure 5-37. Defects and impurities are preserved.

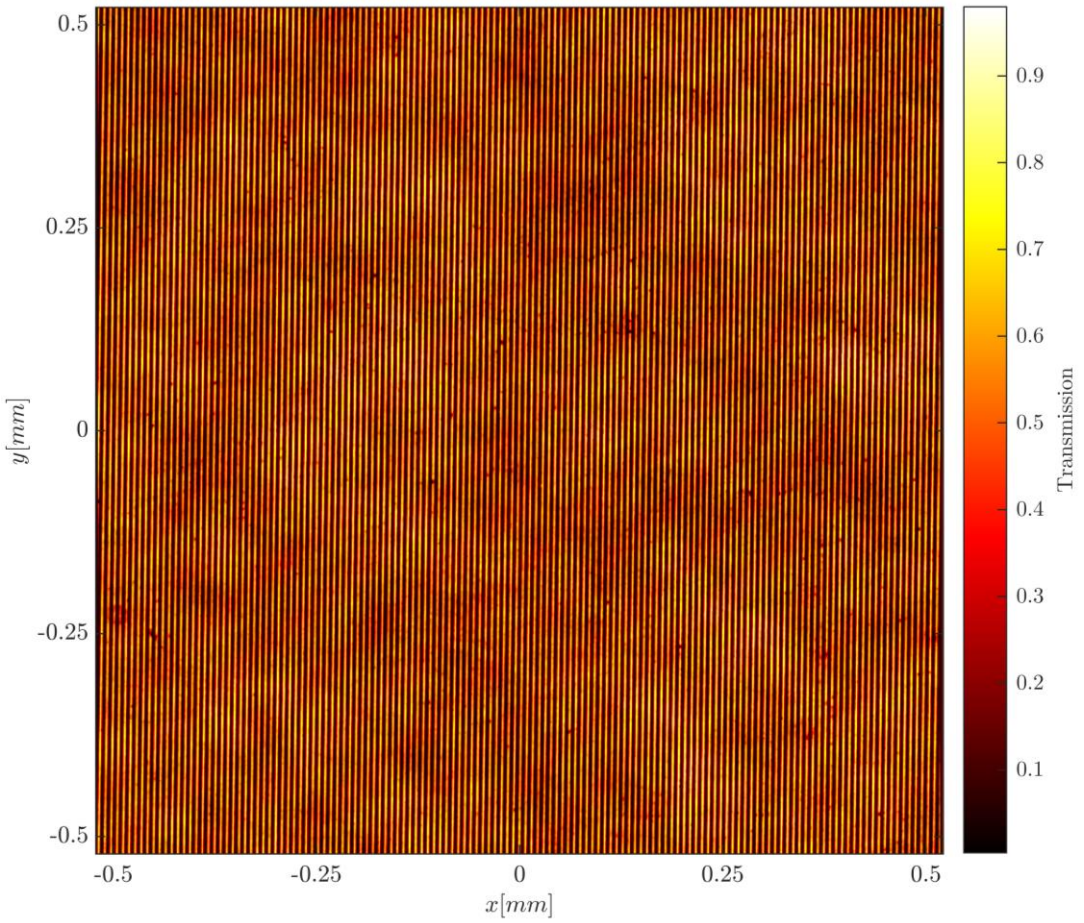


Figure 5-37: Amplitude modulation of the aperture function, derived from the measurements of Figure 5-36. The total field size is $1.0418 \times 1.0418\, mm^2$.

The phase data measured by the WFS is shown in Figure 5-38 a) in units of wavelengths λ, where $\lambda = 632.8\, nm$. Due to the low resolution measurement, the data is interpolated to match the simulation resolution. The interpolation method is again bicubic interpolation. Figure 5-38 b) shows the interpolated wavefront matrix with the phase shift in radians.

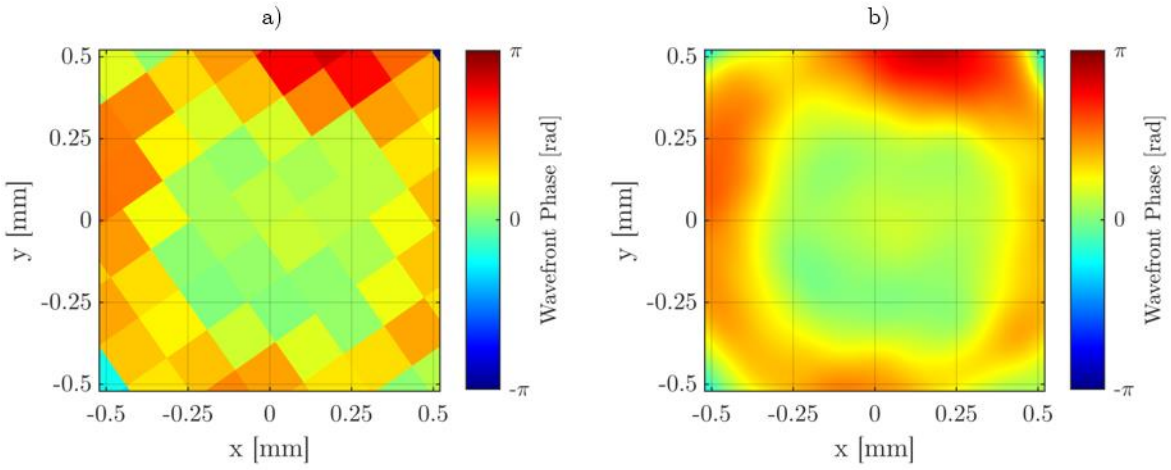


Figure 5-38: a) WFS measurement of the illumination beam phase and b) the bicubic interpolation of the data.

The measured relative intensity distribution of the laser beam is shown in Figure 5-39 a). Because of the low resolution measurement the data is interpolated to match the simulation resolution. The interpolation method is again bicubic interpolation. Figure 5-39 b) shows the interpolated interpolated matrix with the phase shift in radians.

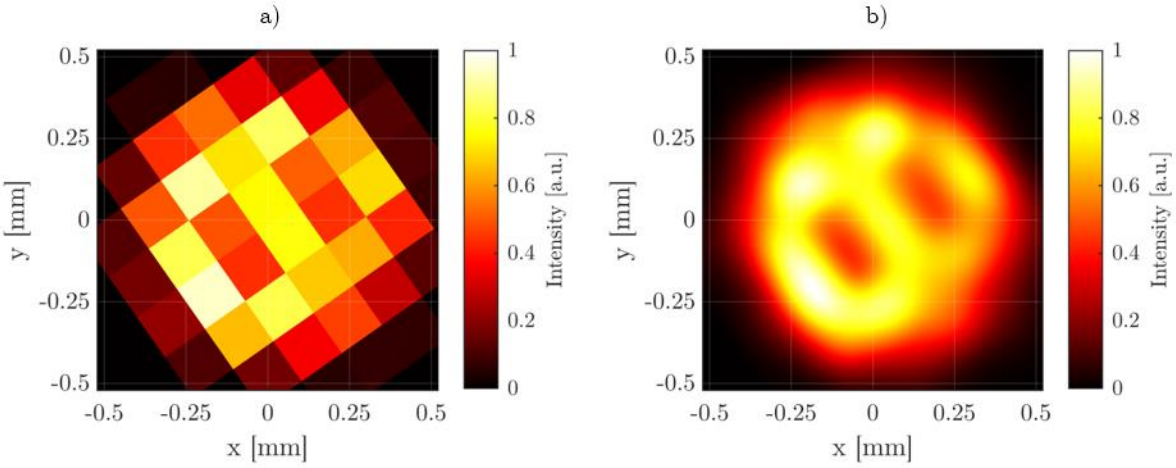


Figure 5-39: a)WFS measurement of the illumination beam intensity distribution in arbitrary units and b) the bicubic interpolation of the data.

The Intensity distribution on the screen has also been measured by a camera, with different exposure times ($1, 5, 10, 50, 100, 500$ and $1000\ \mu s$) added together by their respective exposure time to get a high dynamic range image (HDR) using the MATLAB function *makehdr* [81]. Figure 5-40 shows the real spot on the screen.

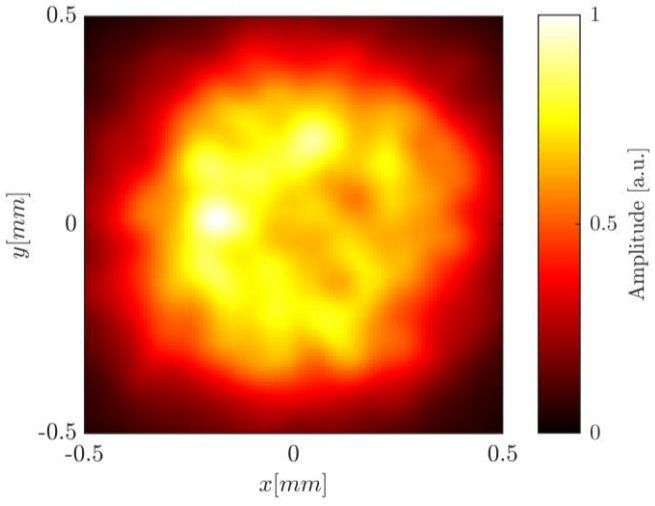


Figure 5-40: HDR image of the laser beam without diffraction grating the the observation screen at distance $z = 30\ mm$.

The intensity image of the diffracted laser beam is also made using the HDR algorithm by MATLAB, with the exposure times of $5, 10, 50, 100, 500$ and $1000\ \mu s$. The image is shown in Figure 5-41. The distance of the first order maxima from the zero order maximum is measured as: $2.52\ mm$ with an angle of $\theta = -3.3°$ to the horizontal axis.

The results of the simulation are shown in Figure 5-42. For the simulation the input beam amplitude and phase have been rotated by $-120°$ to match the WFS detector rotation in the actual setup. The output intensity is normalized so a comparison to Figure 5-41 can be made.

A cross section through the centers of the diffractive orders of Figure 5-41 and Figure 5-42 is shown in Figure 5-43. The calculated position of the $\pm 1^{st}$-orders is indicated by continous lines at $x = \pm 2.54\, mm$. Note that the x-position is not the same in Figure 5-41 and Figure 5-42, because the cross-section is at an angle of $-3.3°$. The position of the diffractive orders in the simulation match the measured and calculated position. The intensity of the simulated first orders is less than the measured intensity. Also, all spots in the simulation are narrower in respect to the measurement.

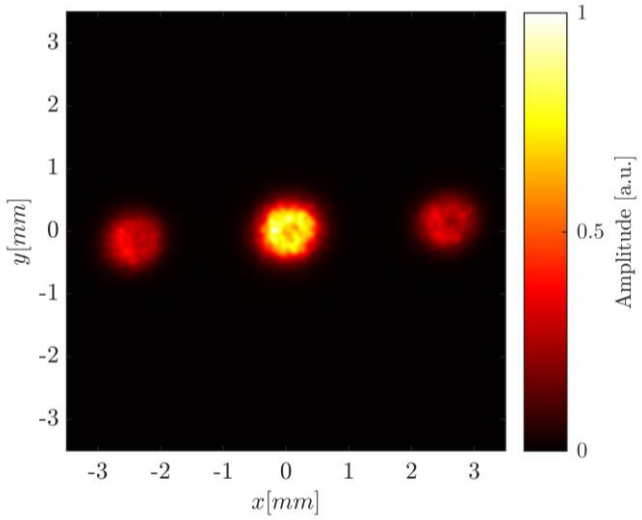


Figure 5-41: HDR image of the diffracted laser beam by the chromium grating at the observation screen $z = 30\, mm$.

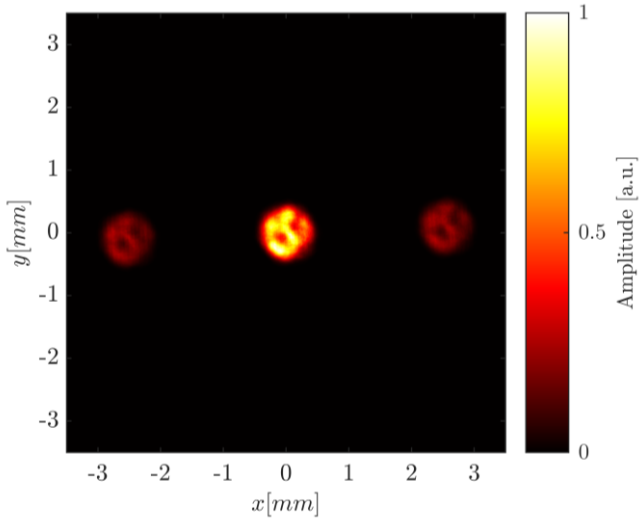


Figure 5-42: Simulation results of a laser beam diffracted by a grating and observed at a screen at a distance of $z = 30\, mm$

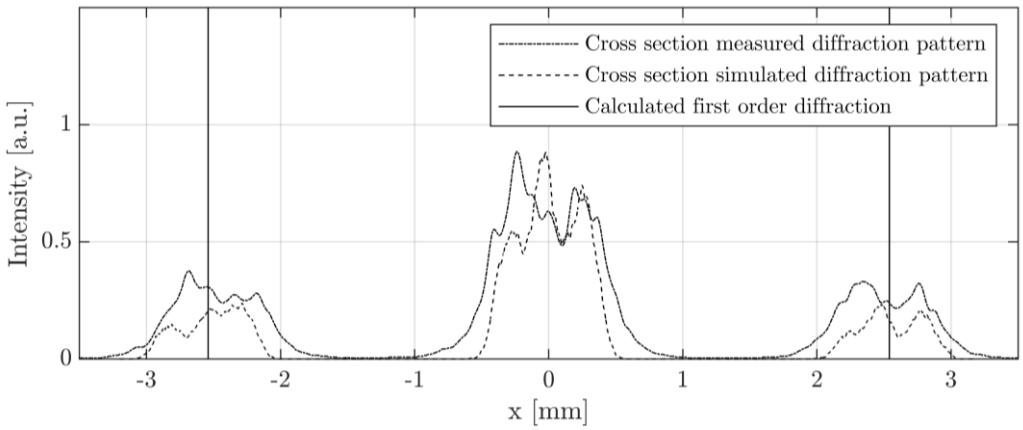


Figure 5-43: Cross section of measured and simulated diffraction pattern. The solid line indicates the calculated first order diffraction distance of $\pm 2.54\,mm$ at the screen.

5.3.3 Discussion

The comparison of a real measured intensity image and a simulation intensity image show that basic principles can be replicated using the BLAS method as a simulation approach. The diffraction by a grating was replicated with a correct diffraction angle. The intensity profiles show similiarities, altough the roation of the the grating with respect to the beam profile could not be calibrated is chosen arbitrarily, as is the location of the laser beam on the grating. Furthermore, the image of the measurement is projected on a white piece of paper with significant surface roughness. Also, the low resolution measurement by the WFS and the bilinear approximation used in the simulation hardly represents the real wavefront..Despite all this approximations and differences in the measurement and simulation, a expected intensity pattern can be simulated. It can be concluded that basic diffraction principles of monochromatic waves can be predicted using the BLAS simulation method developed in this work. By using the measured surface of a diffraction grating including grating defects, surface deviations and dirt, the robustness of the BLAS is considerd to be prooven.

5.4 Scalar Diffraction Simulation of Holographic Surfaces

As the diffractive layers of a D^2NN are in their inherently computer-generated holograms (CGHs), a hologram is calculated in this experimental section to demonstrate the capabilities of the BLAS method. The method used for the computation of the holographic surface is a basic iterative phase retrieval algorithm.

5.4.1 Experimental Setup

First, generating the CGH is explained. The algorithm used is the Gerchberg-Saxton (GS) algorithm [82]. This algorithm is a Fourier transform-based algorithm that uses a lens to reconstruct the hologram in the image plane. Inputs to the algorithm are a source intensity distribution and a target intensity distribution. A schematic illustration of the algorithm is shown in Figure 5-44. To retrieve the phase of the CGH, the target intensity image is initialized with random phase values. An inverse FFT is applied to the complex target and the phase of this transformation is the first iteration of the CGH. In the next iteration, the source intensity distribution is multiplied with the CGH phase. This complex field is transformed using the FFT. This FFT represents the optical field in the image plane and the phase of this field is added to the target intensity distribution. From inverse FFT of this field again the phase is isolated and represents the CGH.

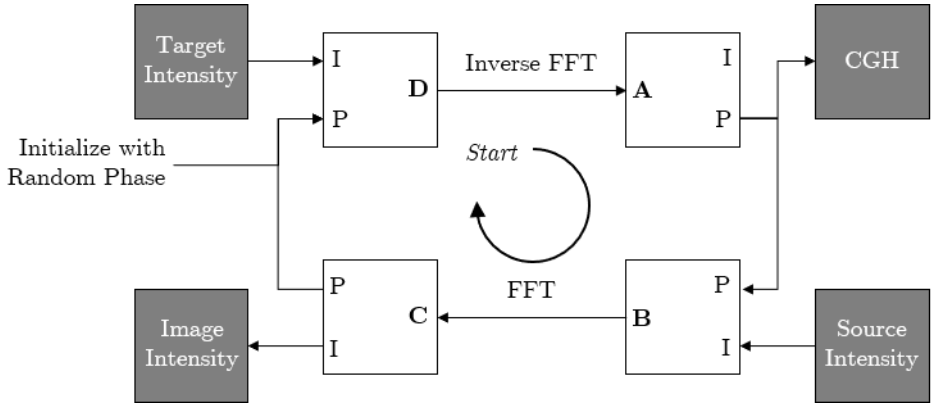


Figure 5-44: Generalized Gerchberg-Saxton phase retrieval algorithm based on [82]. Where A,B,C and D are the variables used in the software implementation. The letters P and I denote the phase and intensity of the complex field, respectivly.

The GS algorithm exploits the circumstance that the transformation of a lens in Fresnel approximation is the Fourier transformation of the object distribution. This concept has already been investigated in the discussion of subsection 5.2.9. However, to reconstruct the image a lens is necessary, or the phase modulation of the lens might be added to the CGH phase. The phase modulation of the lens is again calculated by Code 4. The MATLAB code implementation of the GS algorithm is shown in Code 7. Note the matrix variable notation is the same as in Figure 5-44.

Code 7: Implementation of the Gerchberg-Saxton iterative phase retrieval algorithm

```
function [CGH] = phaseRetrGS(source,target,iterations)
%Gerchberg-Saxton iterative phase retrieval algorithm
%   Input parameters:
%       source: Intensity image of input optical wave
%       target: Intensity image of the traget image
%       iterations: Number of iterative approximations
%   Note that source and target must be the same size.

    % Initilize the target with random phase
    D = abs(target) .* exp(1i.* (rand(size(target)).*2.*pi));
    % Compute complex field of the CGH as starting point
    A = fftshift(ifft2(fftshift(D)));
    % Compute the matrices B,C,D and A iterativly
    for m = 1:iterations
      B = abs(source) .* exp(1i*angle(A));
      C = fftshift(fft2(fftshift(B)));
      D = abs(target) .* exp(1i*angle(C));
      A = fftshift(ifft2(fftshift(D)));
    end
    % Extract the output CGH phase from the complex matrix A
    CGH = angle(A);
```

The computational window size in this experiment is $10 \times 10\,mm^2$, the spatial sampling is $\Delta s = 10\lambda$ and the wavelength is $\lambda = 633\,nm$. The source intensity distribution is chosen to be a gaussian beam profile, which can be described by a normal distribution density function:

$$I(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{r^2}{2\sigma^2}} \tag{5-11}$$

Whereby $r = \sqrt{x^2 + y^2}$ is the radial distance from the center and σ^2 is the variance. For this experiment, the variance is arbitrarily chosen to be $\sigma^2 = 250\,\mu m$. The GS approximation was made using 50 iterations. An intensity image of the source distribution is shown in Figure 5-45 a). The target image for which the CGH is computed for is shown in Figure 5-45 b), which contains binary text and geometrical elements with varying intensity.

The resulting CGH, without any modifications and after 50 iterations, is shown in Figure 5-46 a). Figure 5-46 b) is the CGH if the phase modulation of a lens with the focal length of $f' = 100\,mm$ is added to the phase values and Figure 5-46 c) shows a magnified view of the CGH when an aperture is applied that limits the field to a circular area where the source intensity of Figure 5-45 a) is $> 10^{-3}$. Note that by using the GS algorithm to predict the intensity pattern in the image plane, one automatically applies a Fresnel approximation. This is because the free-space propagation including lens modulation is approximated by a single Fourier transform of the object plane. All images shown in this subsection are outputs of the

GS algorithm and use the Fresnel approximation. The results in the next subsection are calculated using the BLAS algorithm.

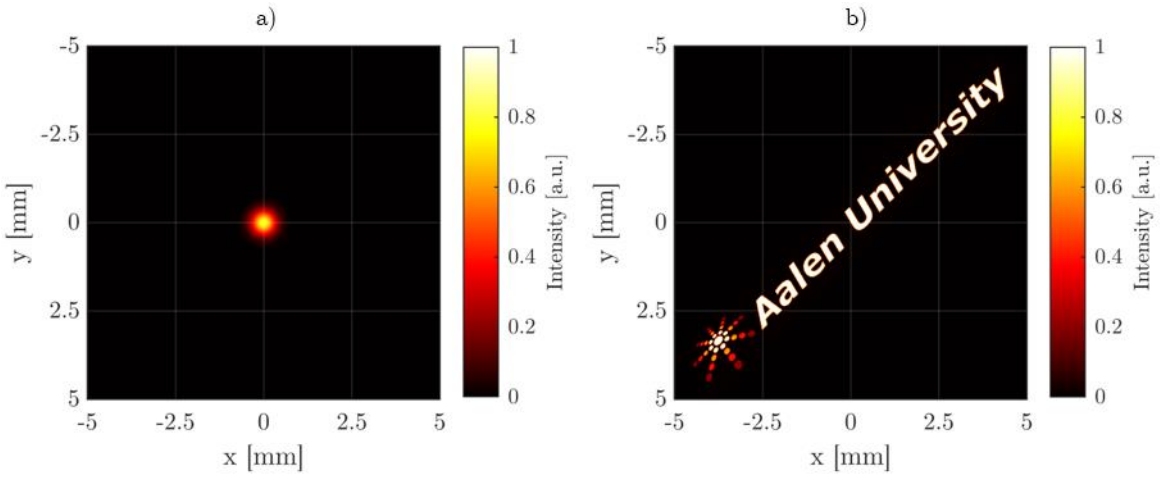


Figure 5-45: GS algorithm gaussian source intensity in a) and the target image is shown in b).

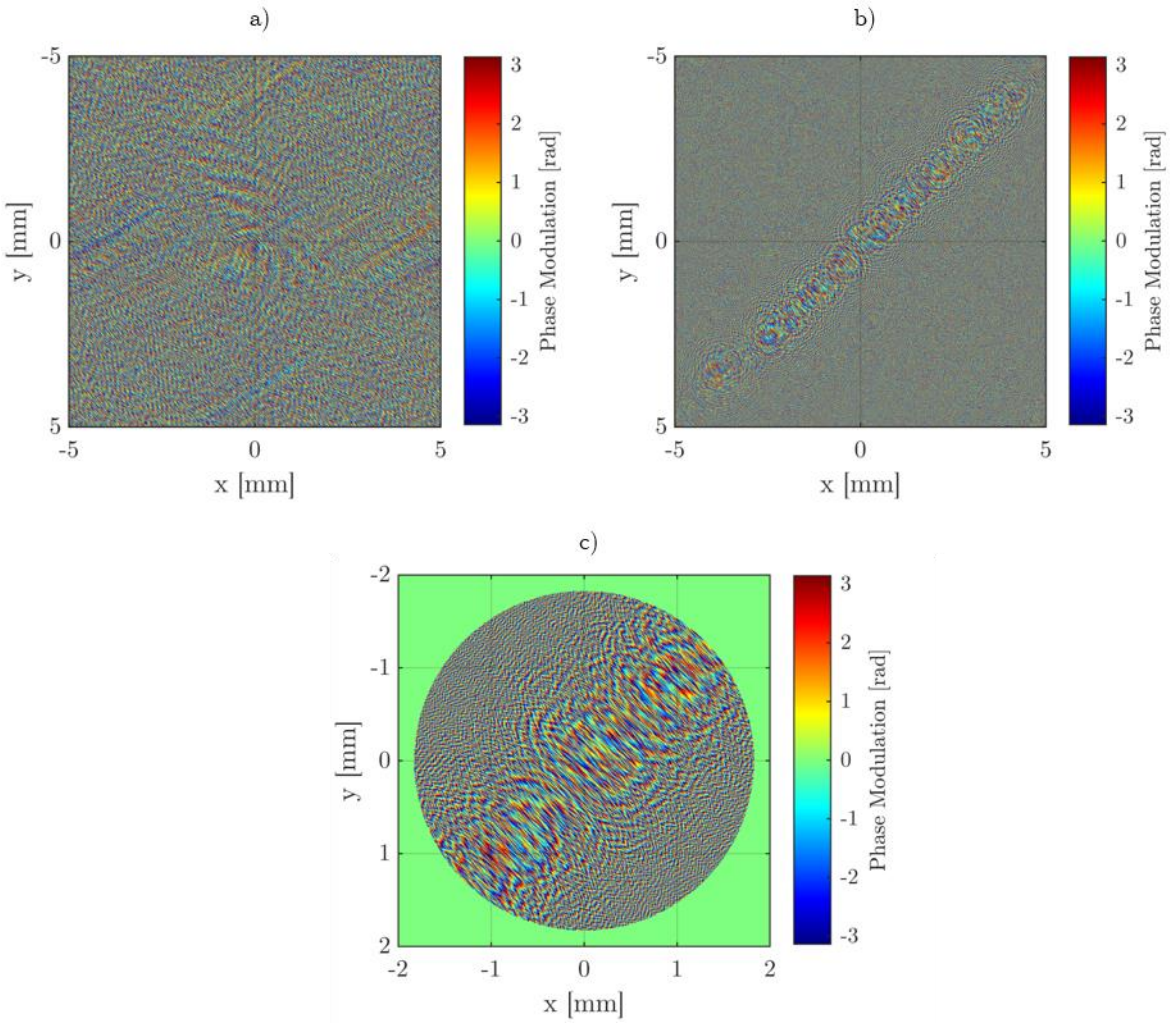


Figure 5-46: Hologram of Figure 5-45 approximated by the GS algorithm in a). Hologram multiplied with a lens modulation function for recunstruction shown in b). A magnified version of b) with a limiting aperture is shown in c), whereby the aperture radius is defined by the radius of the source intensity at which the intensity value is 10^{-3}.

The GS algorithm provides not only a phase map for the object plane, but also a approximated image intensity. The algorithm approximates the lens by a simple Fourier transform, which is identical to a Fresnel approximation of the image plane. This image can be calculated by the absolute squared of the matrix C of Code 7. The result is shown in Figure 5-47 and will serve as a comparison. Figure 5-47 a) shows the image in a linear scale, whereby b) shows the image in a logarithmic scale to visualize the remaining noise around the actual target image due to iterative phase approximation method of the GS algorithm.

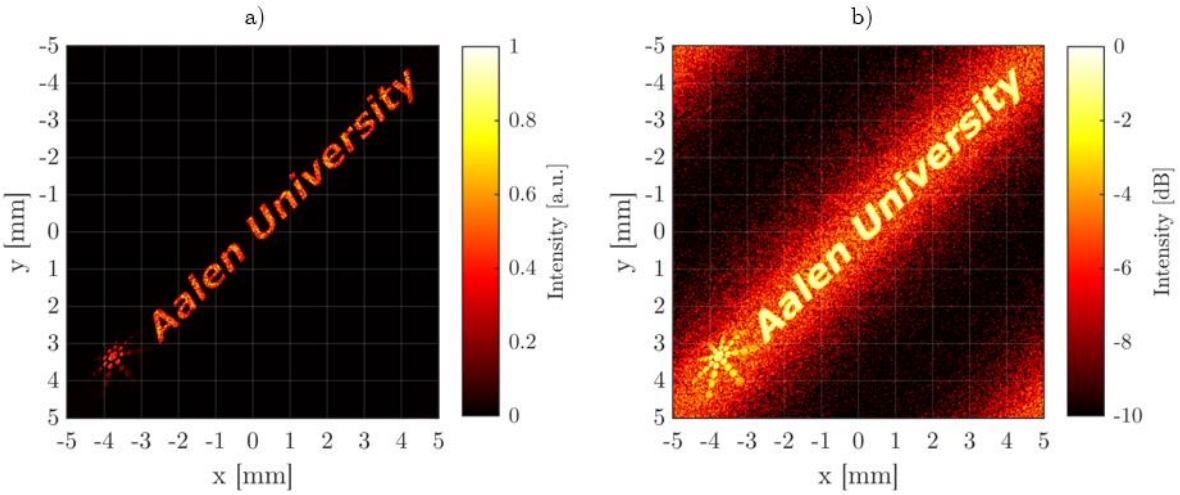


Figure 5-47: Image of the first CGH extracted from the GS algorithm, using a Fresnel approximation. In a) the image in a linear scale is shown and in b) the same image in a logarithmic scale to visualize the noise.

A second binary image will also be calculated for comparison. As target intensity image the USAF chart of Figure 5-20 is used. The sampling, size and distance parameters are the same as the above parameters. The source intensity is also a gaussian profile with the same variance. The CGHs for the USAF chart hologram are shown in Figure 5-48, whereby a) shows the by the GS algorithm computed hologram, b) the hologram multiplied by a lens modulation function with the focal length of $f' = 100\,mm$ and c) is the hologram limited by an aperture with the radius where the source intensity has dropped to 10^{-3}.

The image plane intensity according to the Fresnel lens approximation is shown in Figure 5-49. Whereby in a) the intensity is in a linear scale and in b) the intesity is in a logarithmic scale.

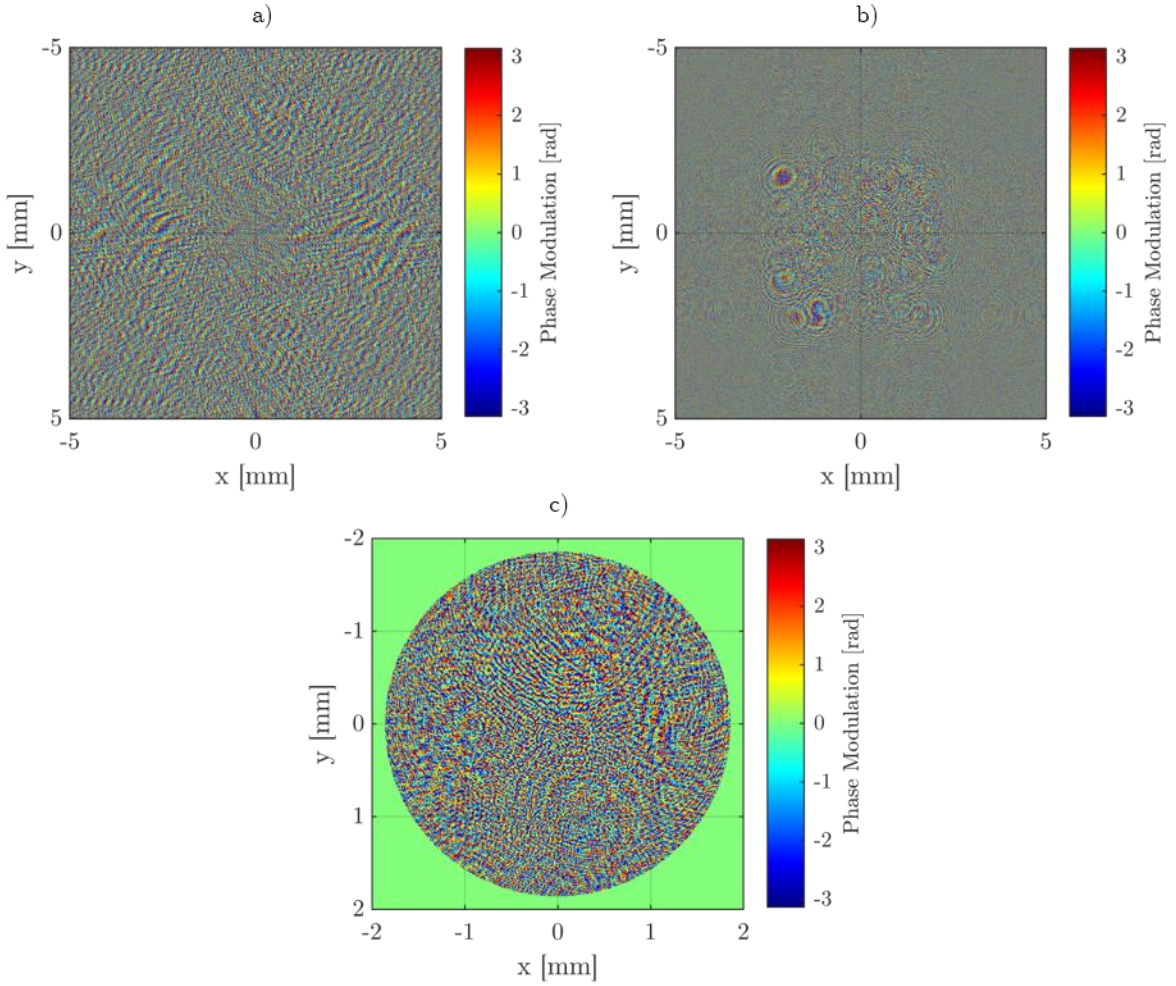


Figure 5-48: Hologram of the USAF chart in Figure 5-20 approximated by the GS algorithm in a). Hologram multiplied with a lens modulation function for reconstruction shown in b). A magnified version of b) with a limiting aperture is shown in c), whereby the aperture radius is defined by the radius of the source intensity at which the intensity value is 10^{-3}.

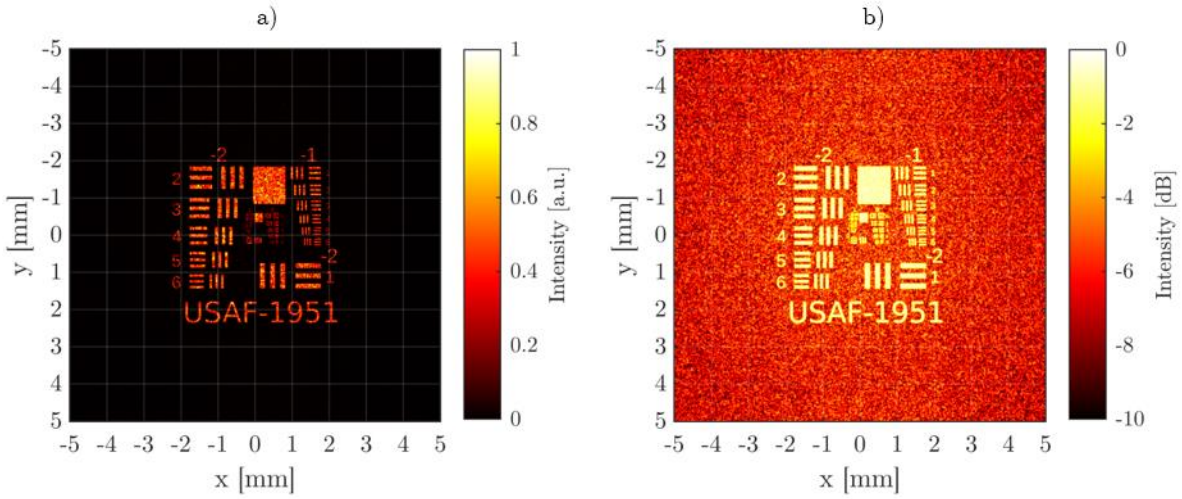


Figure 5-49: Image of the USAF chart CGH extracted from the GS algorithm, using a Fresnel approximation. In a) the image in a linear scale is shown and in b) the same image in a logarithmic scale to visualize the noise.

5.4.2 Results

To achieve a comparable value the MSE and Peak Signal-to-Noise ratio (PSNR) are defined as:

$$MSE = \frac{1}{N}\sum |t - s|^2 \tag{5-12}$$

Whereby N is is the number of pixels in one image, t is the target image and s is the signal image containing the approximation and simulation noise. For the calculation of the MSE the MATLAB function *immse* [83] is used. The PSNR is defined as:

$$PSNR = 10 \cdot \log_{10} \frac{PV^2}{MSE} \tag{5-13}$$

Whereby PV is the peak-to-valley value of the image, which is in case of a normalized image just $PV = 1$.

Propagating the gaussian laser spot in Figure 5-45 a) through the first hologram of Figure 5-46 c) results in the intensity image of Figure 5-50. The intensity image looks identical to the GS approximation of Figure 5-47 a). Similarly, the resulting image in a logarithmic scale shown in Figure 5-51 also looks exactly like the GS approximation of Figure 5-47 b).

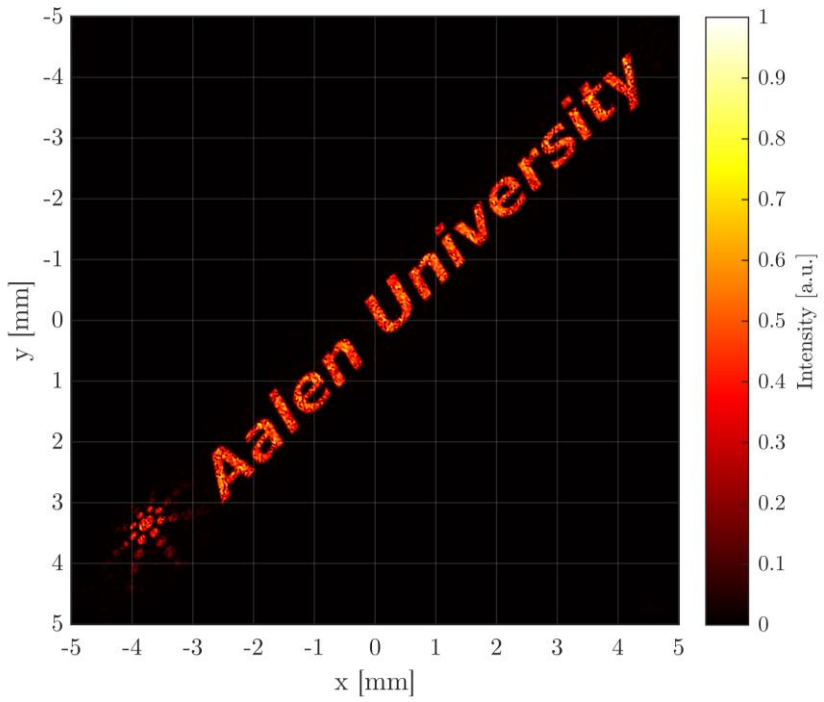


Figure 5-50: Intensity image of a GS CGH of Figure 5-47 using a lens as modulation and the BLAS method for calculation.

The calculated noise values for the BLAS image in Figure 5-50 are: $MSE = 10.5 \cdot 10^{-3}$ and $PSNR = 19.8\,dB$. The values for the approximation of the image in Figure 5-47 are: $MSE = 9.9 \cdot 10^{-3}$ and $PSNR = 20\,dB$.

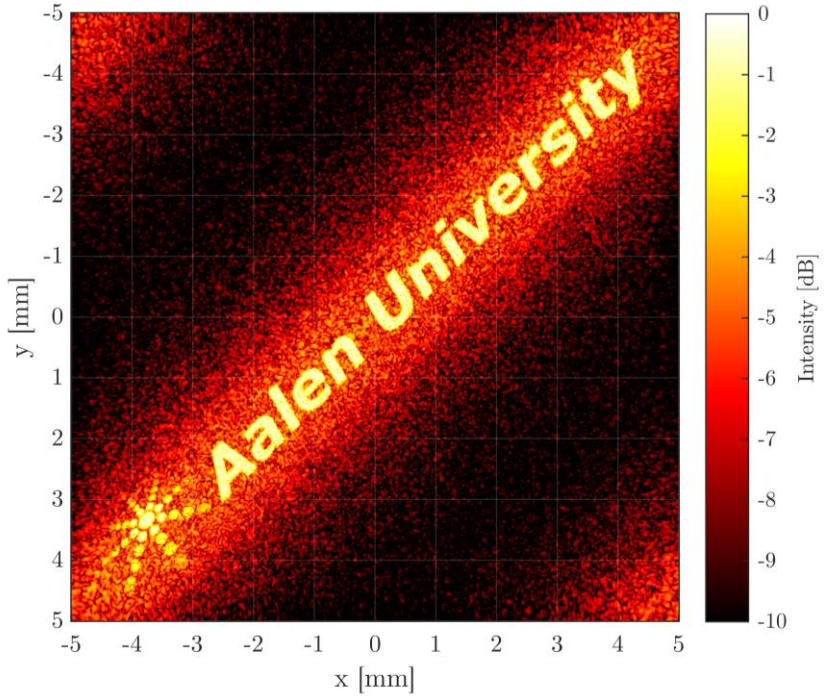


Figure 5-51: Intensity image in a logarithmic scale of a GS CGH of Figure 5-47 using a lens as modulation and the BLAS method for calculation.

The result of the second CGH with BLAS simulation is shown in a linear scale in Figure 5-52 and in logarithmic scale in Figure 5-53. The MSE of the BLAS simulation of Figure 5-52 is $MSE = 18.5 \cdot 10^{-3}$ and the PSNR is $PSNR = 17.3\,dB$. In comparison the image calculated by the GS algorithm has a MSE of $MSE = 18.9 \cdot 10^{-3}$ and the $PSNR = 17.2\,dB$.

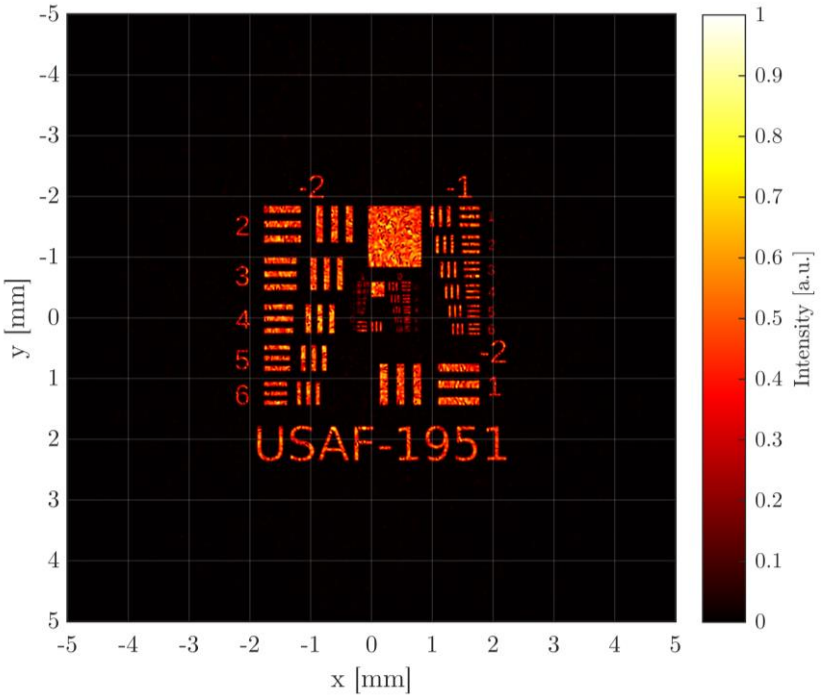


Figure 5-52: Intensity image of a GS CGH of Figure 5-48 using a lens as modulation and the BLAS method for calculation.

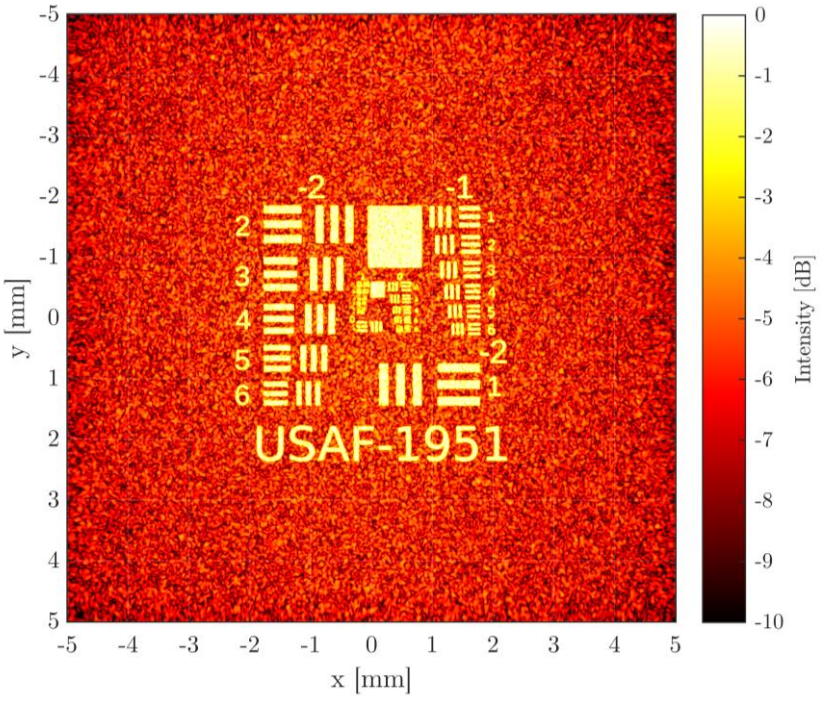


Figure 5-53: Intensity image in a logarithmic scale of a GS CGH of Figure 5-48 using a lens as modulation and the BLAS method for calculation.

5.4.3 Discussion

The experiment in this chapter is done with the concept of the holographic layers in a D^2NN in mind. The ability for the reconstruction of a hologram is fundamental for calculating the forward propagation in a D^2NN. Although the GS algorithm is restricted to use a lens for the

hologram reconstruction, the results of subsection 5.4.2 show that by applying a local phase shift according to the OPD of an ideal lens to the calculated hologram the Fourier transform properties can be included in the hologram itself. Despite of this, the calculated intensity can only be observed at a certain distance, depending on the sampling and numerical aperture of the system. The restriction hereby is again given by the Abbe resolution limit of equation (5-8). Furthermore, the BLAS simulation results match the prediction made by the Fresnel approximation of a lens results quite closely. This might be concluded comparing the PSNR results of subsection 5.4.2. The difference of noise in the images is in the first case approximately $0.2\ dB$ and in the second example $0.1\ dB$, which is objectively relatively small.

However, the use of the GS algorithm is just a means for the proof of concept for holographic reconstruction. Holographic layers optimized by training a D^2NN are not necessarily subject to the numerical aperture limitation, as results of the iterative Fourier approach. Lin, Rivenson et al. [9] proofed that the holograms in a D^2NN work also with smaller distances, as not full neuron connectivity in each layer is demanded. Rather do multiple layers distribute the input energy through several network layers and thus provide interconnectivity. By changing the diffractive layer distance, the weighting matrix is uniformly altered. When shorter distances are chosen, it may be necessary to add more layers to provide a constant degree of connectivity. For applications where only local connectivity is demanded, short layer distances are completely acceptable, as Lin, Rivenson et al. showed in their imaging application [9], where only local image noise or image errors are corrected.

From the theoretical standpoint, the angular spectrum method is accurate in near-field given the approximations of by the Fresnel-Kirchhoff diffraction integral of subsection 4.1.3, whereby considering the aperture and the limitation of plane input and output planes of the Rayleigh-Sommerfeld diffraction integral of subsection 4.1.4. The BLAS calculation of the Fresnel regime must therefore be given and has been proven by the experiment of this section. The conclusion of this experiment is that the BLAS method is suitable for calculating the forward propagation model of a D^2NN, due to its capability for holographic reconstruction proven here.

6 Summary

In this thesis an algorithm for training a diffractive deep neural network has been derived and tested. Therefore, an introduction to basic neural networks and the training algorithm has been given in chapter 2. A mathematical description of an artificial neuron and how they are connected in a shallow neural network has been presented. The explained training algorithm for the network training is the stochastic gradient descent method. To emphasize the importance of non-linear elements in neural networks a simple example was given. Furthermore, the basic principle of convolutional neural networks was explained.

In the third chapter the current state of the art in optical computing has been reviewed and the concept of diffractive neural networks based on free-space propagation has been further explored. Therefore, a mathematical model of D^2NNs and a backpropagation algorithm has been derived using the model for classical artificial neural networks and recent publications in the field [8, 9, 45–48, 53]. An important conclusion is that, as for classical NNs the network error can be backpropagated through the network to calculate the network parameter gradients. Zhou, Fang et al. even showed that if the error field is treated as an optical field, the gradients to minimize the network error can exist physically. Based on the examined properties the limitations of current methods have been investigated. Current implementations of D^2NNs [9, 46, 48] use terahertz wavelengths as diffractive elements can be easily manufactured.

The method for calculating the forward propagation used is the Rayleigh-Sommerfeld integral, which has the limitation of reasonable computational effort for actual optical wavelengths in the VIS and NIR range. In section 3.3 two hypotheses have been derived based on these current limitations and one further hypothesis was made. If Hypothesis 1 to Hypothesis 3 can be fulfilled, the developed methods build the foundation for calculating D^2NNs and convolutional D^2NNs. For Hypothesis 3 a conceptual design has been proposed that has been theoretically tested in this thesis.

The method that promises to provide the computational efficiency needed is the angular spectrum method. Chapter 4 is an in-depth derivation of the AS method. Because diffraction simulations are not universally applicable, a profound understanding of the Rayleigh-Sommerfeld diffraction simulation and the AS method is required to minimize the computational effort needed. In section 3.3 the necessity for fast calculation has been demonstrated. To allow diffraction simulation the AS method was refined by a bandwidth limit

based on the work of Yu, Xiahui et al. [67]. The sampling conditions have been extensively investigated to derive a set of sampling rules in subsection 4.2.3. This set of sampling rules is tested and examined in the experimental chapter. An addition to free-space diffraction simulation, the scalar modulation of an optical field by surfaces and optical elements was presented in section 4.3. Furthermore, an algorithmic implementation of the BLAS method and a standalone diffraction simulation by using MATLAB were developed in section 4.4 and 4.5 respectively.

In the experimental part four main applications of the developed algorithm have been investigated. In the first part the computational speed was investigated using a common consumer CPU for calculation. The duration of a wide range of computation field sizes have been measured and were compared against a straight-forward calculation of the RS integral and a direct integration method. The results of subsection 5.1.2 demonstrate that the advantage of the BLAS algorithm is evident. Depending on the computational field size, BLAS method is magnitudes faster than the compared methods. With increasing field size, the BLAS method gains even more advantage. Also, computational segments of the BLAS algorithm have been measured individually to estimate the calculation time for sequential propagation layers with equal distance. An example estimation for calculating a D^2NN with five layers shows that the difference in computation time is tremendous. When using the BLAS method, the pure propagation calculation time was estimated to approximately $58\,h$ whereas the direct RS calculation was estimated to approximately $24\,years$. In addition, the hardware requirements in terms of memory usage have been measured. The results in subsection 5.1.2 show, that when limiting the value precision to single precision float complex numbers the memory usage is reasonable and can be handled by common PC setups.

The second experimental section theoretically explores the possibility of a new kind of optical convolutional layer. The concept was first introduced in section 3.3 and has the potential to provide a new way of abstraction in D^2NNs. To the best of my knowledge, this is the first suggestion of a multi-kernel convolutional layer that is conceptual capable of a mean pooling function. A similar concept has been proposed by Yan, Wu et al. [47] in 2019 but with different functionality, as they only use the convolutional properties of a 4-f setup for image filtering and salient object highlighting. The experimental section 5.2 was divided into three subsections.

The first subsections 0 to 5.2.3 proved the application of the BLAS algorithm to basic optical components like lenses and grating generate expected results.

The second subsections 0 to 5.2.9 then examined the sampling condition of a symmetric telescopic 4-f system. Because the objective of a convolutional unit is to modulate the optical field in the Fourier plane, the validity of the sampling condition is reviewed. The main result of this experiment is that the validity of the frequency image is subject not only to the sampling conditions derived in 4.2.3 but also to an inverse Abbe resolution limit.

The third part in subsections 5.2.10 to 5.2.12 put all components together and illustrated that from a conceptual point of view a multi-kernel convolution unit produces multiple images with multiple frequency images in one Fourier plane. The multiple images vary in intensity predictively depending on the diffraction efficiency, but it was found that each wavefront phase is a superposition of the phase with respect to the physical location in the focal plane and specific offset of the corresponding image path. Concluding that the calculation of a multi-kernel convolution cannot be split into several paths, but rather must be calculated as a whole, which requires a fast calculation method.

The third experiment in section 5.3 demonstrated the capability to reconstruct real experimental conditions by using measured input data and compared the simulation results to measurements. As test setup a diffraction grating illuminated by a circular aperture was chosen.

In the last section 5.4 a holographic element was computed using the Gerchberg-Saxton algorithm. The light propagation through the hologram was computed using the BLAS method and the resulting image was compared to the Fresnel approximation of the image, which is a side product of the Gerchberg-Saxton algorithm. The results are almost identical showing only slight difference in noise. The motivation for this experiment was to show that the derived BLAS algorithm is suitable for calculating not only convolutional units but also feed forward layers in D^2NNS.

Although the actual training of D^2NN is out of scope for this work, the derived and developed methods enable the calculation of D^2NNs operating at VIS and NIR wavelengths. With this "toolbox" of algorithms and guidelines for sampling, the optical forward propagation and error backpropagation might be calculated. Furthermore, these algorithms and methods can be applied to any scalar monochromatic diffraction problem. This confirms Hypothesis 1 that stated*: With a more efficient algorithm to calculate the diffracted optical field, the forward propagation for training a D^2NN for NIR and VIS operation wavelength, can be calculated in a macroscopic scale on a common desktop PC.* The experiments often were oversampled, especially in 5.3, so that can be concluded that Hypothesis 2: *The algorithm of Hypothesis 1 is*

fast enough to allow subsampling of neuron structures to include surface deviations into the training of a D^2NN and still remain reasonably fast when using NIR or VIS wavelengths. can also be confirmed. Additionally, the investigated concept of multi-kernel convolutional diffractive neural network promises a large protentional to achieve more sophisticated image recognition capabilities. This means that Hypothesis 3: *A physically accurate forward propagation model can reproduce a multiple kernel convolution in one computational task, according to the concept of Figure 3-16, for the use of a Fourier-space convolutional deep diffractive neural network* basically is also confirmed, but a physical implementation and actual evidence of the computational benefits have yet to be demonstrated.

7 Outlook

As this thesis issues mostly theoretical sampling problems of the BLAS method, the logical next step is a physical implementation of one multi-kernel convolution unit to verify the results of this work. The optical power throughput of the multi-kernel design must be evaluated by measuring the SNR of each image depending on number of kernels implemented.

An assumption made in this work is, that modulating the Fourier plane image in amplitude and phase results in predictable results in the image plane. The concept of frequency image filtering is well known and widely applied in respect to amplitude modulation. A filtering method that might be used to verify the phase filtering capabilities is the application of airy spiral filters [84] which use complex valued modulation function for image microscopic enhancement.

In respect to modifications to the BLAS exist several approaches which could improve the algorithms' performance and simulation accuracy. One modification that might be implemented is the improvement of the bandwidth limit in the free-space transfer function of the BLAS method. Falaggis, Kozacki et al. showed that by applying smooth filtering to the transfer function the simulation accuracy increases and aliases appear at much higher distances in the observation plane [85]. A further proposal for increasing the accuracy for high numerical BLAS simulations was made by Falaggis, Kozacki et al [86], which should result in further improvements.

Two major limitations exist in all mentioned methods of AS diffraction calculations. The first limitation is that the observation window must be the same size as the aperture window, and both are centred around one perpendicular axis. When one wants to observe the optical field far away from this optical axis, the computational window size must be increased so that the observed area falls into the calculated field. This results in a large number of unnecessary calculations. The second limitation is that the aperture plane and observation plane must be parallel. If an experimental setup requires tilted plane and folded beam paths, the number of fields to calculate increases because the tilted plane must then be additionally sampled in the z-direction as well. Both limitations have been resolved by Matsushima et al. which originally proposed the BLAS method. The method for calculating off-axis windows in the observation plane can be found in [87] and the method for calculating tilted planes in [88]. With these tools any physical setup should be covered, except for non-planar calculation planes.

A crucial further development for this thesis' work is the actual implementation of a D^2NN training algorithm. In the chapters 2, 3 and 4 all essential mathematical concepts were presented. By using these concepts, a complete neural network setup should be possible. First investigations proved that this task is highly non-trivial due to the vast algorithmic complexity and cumbersome debugging due to high calculation times. Nevertheless, further developments for a complete implementation of a D^2NN training and simulation program is already ongoing. The implementation success is important to enable research on networks convergence when using the BLAS method for the forward propagation and error backpropagation, as well as further research regarding to the proof of concept for the multi-kernel convolutional unit.

For the long-term perspective, the goal is to realize a transmissive or reflective D^2NN for wavelengths in the VIS and NIR range by using lithographic manufacturing processes and nano-imprint replication. Additionally, the concept of non-monochromatic D^2NNs is a field worthy of further research. Basic investigations have already been made by Lou, Mengu et al. [89]. Also, the Rayleigh-Sommerfeld integral, from which the free-space transfer function is derived, can be generalized to a non-monochromatic case [32, pp 53,54]. This allows to calculate the diffraction pattern for polychromatic coherent illumination, which is the next step for an all-optical intelligent sensor.

To summarize this outlook, the experimental results should be compared with further real physical implementations. Also, further improvements to the BLAS method enable more generalized physical configurations. The most important step in the near future is the successful training of a BLAS-based D^2NN to verify that a multi-kernel convolution layer can improve the image classification capabilities. As long-term goals, a fully functional setup must be manufactured.

8 Bibliography

1. Schaller, R.R.: Moore's law: past, present and future. IEEE spectrum 34, 52–59 (1997)
2. Powell, J.R.: The quantum limit to Moore's law. Proc. IEEE 96, 1247–1248 (2008)
3. Kumar, S.: Fundamental Limits to Moore's Law. http://arxiv.org/pdf/1511.05956v1 (2015)
4. IBM Quantum - Quantum Computing at IBM. https://www.ibm.com/quantum-computing/quantum-computing-at-ibm/ (2021). Accessed 11 January 2021
5. Touch, J., Badawy, A.-H., Sorger, V.J.: Optical computing. Nanophotonics (2017). https://doi.org/10.1515/nanoph-2016-0185
6. Aggarwal, C.C.: Neural networks and deep learning. A textbook / Charu C. Aggarwal. Springer, Cham, Switzerland (2018)
7. Web of Science [v.5.35] - Web of Science Core Collection Basic Search. http://apps.webofknowledge.com/WOS_GeneralSearch_input.do?product=WOS&search_mode=GeneralSearch&SID=E53BhcZlJf4Lsv92JDm&preferencesSaved= (2021). Accessed 11 January 2021
8. Marinis, L. de, Cococcioni, M., Castoldi, P., Andriolli, N.: Photonic Neural Networks: A Survey. IEEE Access (2019). https://doi.org/10.1109/ACCESS.2019.2957245
9. Lin, X., Rivenson, Y., Yardimci, N.T., Veli, M., Luo, Y., Jarrahi, M., Ozcan, A.: All-optical machine learning using diffractive deep neural networks. Science (2018). https://doi.org/10.1126/science.aat8084
10. Micheal A. Nielsen: Neural Networks and Deep Learning. Determination Press (2015)
11. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A.: Fourier Neural Operator for Parametric Partial Differential Equations. http://arxiv.org/pdf/2010.08895v1 (2020)
12. Da Silva, I.N., Spatti, D.H., Andrade Flauzino, R., Liboni, L.H.B., Reis Alves, S.F.d.: Artificial neural networks. A practical course / Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, Silas Franco dos Reis Alves. Springer, Switzerland (2016)

13. Zissis, D., Xidias, E.K., Lekkas, D.: A cloud based architecture capable of perceiving and predicting multiple vessel behaviour. Applied Soft Computing (2015). https://doi.org/10.1016/j.asoc.2015.07.002

14. Sengupta, N., Sahidullah, M., Saha, G.: Lung sound classification using cepstral-based statistical features. Computers in biology and medicine (2016). https://doi.org/10.1016/j.compbiomed.2016.05.013

15. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. http://arxiv.org/pdf/1604.00449v1 (2016)

16. Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. In: Palm, G., Aertsen, A. (eds.) Brain Theory, pp. 245–248. Springer Berlin Heidelberg, Berlin, Heidelberg (1986)

17. Georgiou, G.M., Koutsougeras, C.: Complex domain backpropagation. IEEE Trans. Circuits Syst. II (1992). https://doi.org/10.1109/82.142037

18. Gedeon, T., Wong, K.W., Lee, M.: Neural Information Processing, vol. 1143. Springer International Publishing, Cham (2019)

19. Braga-Neto, U.d.M.: Fundamentals of pattern recognition and machine learning. Springer, Cham, Switzerland (2020)

20. Brownlee, J.: How to Choose Loss Functions When Training Deep Learning Neural Networks. Machine Learning Mastery, 29 January 2019. https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/. Accessed 11 December 2020

21. Werbos, P.J.: Applications of Advances in Nonlinear Sensitivity Analysis. In: Proceedings of the 10th IFIP Conference, 31.8 - 4.9, NYC, pp. 762–770 (1981)

22. Rumelhart, D.E., McClelland, J.L.: An interactive activation model of context effects in letter perception: II. The contextual enhancement effect and some tests and extensions of the model. Psychological Review (1982). https://doi.org/10.1037/0033-295X.89.1.60

23. Parker, D.B.: Learning Logic. Technical Report TR-47. Massachusetts Institute of Technology, Cambridge, MA, USA (1985)

24. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature (1986). https://doi.org/10.1038/323533a0

25. Peter Sadowsk: Notes on Backpropagation. https://www.ics.uci.edu/~pjsadows/notes.pdf. Accessed 12 April 2020

26. Minsky, M.L., Papert, S.: Perceptrons. An introduction to computational geometry. Massachusetts Institute of Technology, Cambridge, Mass. (1990, 1969)

27. 2-D convolution - MATLAB conv2 - MathWorks Deutschland. https://de.mathworks.com/help/matlab/ref/conv2.html (2020). Accessed 6 December 2020

28. Abraham, S.: Image Manipulation. Filters and Convolutions. https://www.cs.utexas.edu/~theshark/courses/cs324e/lectures/cs324e-6.pdf (2020). Accessed 12 July 2020

29. Kanopoulos, N., Vasanthavada, N., Baker, R.L.: Design of an image edge detection filter using the Sobel operator. IEEE J. Solid-State Circuits (1988). https://doi.org/10.1109/4.996

30. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE (1998). https://doi.org/10.1109/5.726791

31. Bouvrie, J.: Notes on Convolutional Neural Networks. http://cogprints.org/5869/1/cnn_tutorial.pdf (2006). Accessed 12 July 2020

32. Goodman, J.W.: Introduction to Fourier optics, 3rd edn. Roberts, Englewood (Colo.) (op. 2005)

33. Farhat, N.H., Psaltis, D., Prata, A., Paek, E.: Optical implementation of the Hopfield model. Applied optics (1985). https://doi.org/10.1364/AO.24.001469

34. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America (1982). https://doi.org/10.1073/pnas.79.8.2554

35. Ohta, J., Takahashi, M., Nitta, Y., Tai, S., Mitsunaga, K., Kjuma, K.: A new approach to a GaAs/AlGaAs optical neurochip with three layered structure. In: Proc. IJCNN International Joint Conference on Neural Networks, pp. 477–482 (1989)

36. Kuratomi, Y., Takimoto, A., Akiyama, K., Ogawa, H.: Optical neural network using vector-feature extraction. Applied optics (1993). https://doi.org/10.1364/AO.32.005750

37. Psaltis, D., Brady, D., Gu, X.-G., Lin, S.: Holography in artificial neural networks. In: Landmark Papers On Photorefractive Nonlinear Optics, pp. 541–546. World Scientific (1995)

38. Barrett, M.D., Chiaverini, J., Schaetz, T., Britton, J., Itano, W.M., Jost, J.D., Knill, E., Langer, C., Leibfried, D., Ozeri, R., Wineland, D.J.: Deterministic quantum teleportation of atomic qubits. Nature (2004). https://doi.org/10.1038/nature02608

39. Agrell, E., Karlsson, M., Chraplyvy, A.R., Richardson, D.J., Krummrich, P.M., Winzer, P., Roberts, K., Fischer, J.K., Savory, S.J., Eggleton, B.J., Secondini, M., Kschischang, F.R., Lord, A., Prat, J., Tomkos, I., Bowers, J.E., Srinivasan, S., Brandt-Pearce, M., Gisin, N.: Roadmap of optical communications. J. Opt. (2016). https://doi.org/10.1088/2040-8978/18/6/063002

40. Tait, A.N., Lima, T.F. de, Zhou, E., Wu, A.X., Nahmias, M.A., Shastri, B.J., Prucnal, P.R.: Neuromorphic photonic networks using silicon photonic weight banks. Scientific reports (2017). https://doi.org/10.1038/s41598-017-07754-z

41. Shen, Y., Harris, N.C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D., Soljačić, M.: Deep learning with coherent nanophotonic circuits. Nature Photon (2017). https://doi.org/10.1038/nphoton.2017.93

42. Shen, Y., Bai, Y.: Statistical Computing with Integrated Photonics System. In: 2019 24th OptoElectronics and Communications Conference (OECC) and 2019 International Conference on Photonics in Switching and Computing (PSC). 2019 24th OptoElectronics and Communications Conference (OECC) and 2019 International Conference on Photonics in Switching and Computing (PSC), Fukuoka, Japan, 07/07/2019 - 11/07/2019, p. 1. IEEE (2019 - 2019). https://doi.org/10.23919/PS.2019.8817791

43. Bao, Q., Zhang, H., Ni, Z., Wang, Y., Polavarapu, L., Shen, Z., Xu, Q.-H., Tang, D., Loh, K.P.: Monolayer graphene as a saturable absorber in a mode-locked laser. Nano Res. (2011). https://doi.org/10.1007/s12274-010-0082-9

44. Nozaki, K., Tanabe, T., Shinya, A., Matsuo, S., Sato, T., Taniyama, H., Notomi, M.: Sub-femtojoule all-optical switching using a photonic-crystal nanocavity. Nature Photon (2010). https://doi.org/10.1038/nphoton.2010.89

45. Zuo, Y., Li, B., Zhao, Y., Jiang, Y., Chen, Y.-C., Chen, P., Jo, G.-B., Liu, J., Du, S.: All-optical neural network with nonlinear activation functions. Optica 6, 1132–1137 (2019)

46. Mengu, D., Luo, Y., Rivenson, Y., Ozcan, A.: Analysis of Diffractive Optical Neural Networks and Their Integration with Electronic Neural Networks. IEEE journal of selected topics in quantum electronics : a publication of the IEEE Lasers and Electro-optics Society (2020). https://doi.org/10.1109/JSTQE.2019.2921376

47. Yan, T., Wu, J., Zhou, T., Xie, H., Xu, F., Fan, J., Fang, L., Lin, X., Dai, Q.: Fourier-space Diffractive Deep Neural Network. Physical review letters (2019). https://doi.org/10.1103/PhysRevLett.123.023901

48. Zhou, T., Fang, L., Yan, T., Wu, J., Li, Y., Fan, J., Wu, H., Lin, X., Dai, Q.: In situ optical backpropagation training of diffractive optical neural networks. Photon. Res. (2020). https://doi.org/10.1364/PRJ.389553

49. Hernández-Hernández, E., Domínguez-Cruz, R., Iturbe-Castillo, M.D., Ramos-García, R.: Optical Characterization of SBN:60 and its Application as an Electric-Field Tunable Photorefractive Filter. In: Andersen, P.E. (ed.) Advances in photorefractive materials, effects and devices. Advances in Photorefractive Materials, Effects and Devices, Elsinore, AD17. OSA, Washington, D.C. (1999). https://doi.org/10.1364/APMED.1999.AD17

50. Hirose, A.: Complex-valued neural networks, 2nd edn. Studies in computational intelligence, 1860-949X, v. 400. Springer, Heidelberg (2012)

51. Hirose, A.: Complex-valued neural networks: The merits and their origins. In: 2009 International joint conference on neural networks, pp. 1237–1244 (2009)

52. Lin, X., Rivenson, Y., Yardimci, N.T., Veli, M., Luo, Y., Jarrahi, M., Ozcan, A.: All-optical machine learning using diffractive deep neural networks. Science (2018). https://doi.org/10.1126/science.aat8084

53. Hughes, T.W., Minkov, M., Shi, Y., Fan, S.: Training of photonic neural networks through in situ backpropagation and gradient measurement. Optica (2018). https://doi.org/10.1364/OPTICA.5.000864

54. Saleh, B.E.A., Teich, M.C.: Fundamentals of photonics, 2nd edn. Wiley series in pure and applies optics. Wiley; [Chichester : John Wiley, Hoboken, N.J. (2007)

55. Busch, S.F., Weidenbach, M., Fey, M., Schäfer, F., Probst, T., Koch, M.: Optical Properties of 3D Printable Plastics in the THz Regime and their Application for 3D Printed THz Optics. J Infrared Milli Terahz Waves (2014). https://doi.org/10.1007/s10762-014-0113-9

56. Mehrabkhani, S., Schneider, T.: Is the Rayleigh-Sommerfeld diffraction always an exact reference for high speed diffraction algorithms? Optics express (2017). https://doi.org/10.1364/OE.25.030229

57. Shi, J., Chen, M., Wei, D., Hu, C., Luo, J., Wang, H., Zhang, X., Xie, C.: Anti-noise diffractive neural network for constructing an intelligent imaging detector array. Opt. Express (2020). https://doi.org/10.1364/oe.405798

58. Dr. Rüdiger Paschotta: Optical Intensity. RP-Photonics. https://www.rp-photonics.com/optical_intensity.html. Accessed 11 November 2020

59. Appendix B: The Scalar Theory of Diffraction. In: Kress, B.C., Meyrueis, P. (eds.) Applied Digital Optics, pp. 587–595. John Wiley & Sons, Ltd, Chichester, UK (2009)

60. Cooper, I.J., Sheppard, C., Sharma, M.: Numerical integration of diffraction integrals for a circular aperture. Optik (2002). https://doi.org/10.1078/0030-4026-00170

61. Matsushima, K., Shimobaba, T.: Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. Optics express (2009). https://doi.org/10.1364/OE.17.019662

62. Shen, F., Wang, A.: Fast-Fourier-transform based numerical integration method for the Rayleigh-Sommerfeld diffraction formula. Applied optics (2006). https://doi.org/10.1364/AO.45.001102

63. Sherman, G.C.: Application of the convolution theorem to Rayleigh's integral formulas. Journal of the Optical Society of America (1967). https://doi.org/10.1364/JOSA.57.000546

64. Cooley, J.W., Tukey, J.W.: An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematics of Computation (1965). https://doi.org/10.2307/2003354

65. Shannon, C.E.: Communication in the Presence of Noise. Proc. IRE (1949). https://doi.org/10.1109/jrproc.1949.232969

66. Shimobaba, T., Takahashi, T., Yamamoto, Y., Nishitsuji, T., Shiraki, A., Hoshikawa, N., Kakue, T., Ito, T.: Efficient diffraction calculations using implicit convolution. OSA Continuum (2018). https://doi.org/10.1364/OSAC.1.000642

67. Yu, X., Xiahui, T., Yingxiong, Q., Hao, P., Wei, W.: Band-limited angular spectrum numerical propagation method with selective scaling of observation window size and sample number. Journal of the Optical Society of America. A, Optics, image science, and vision (2012). https://doi.org/10.1364/JOSAA.29.002415

68. Horn, R.A., Johnson, C.R.: Matrix analysis, 2nd edn. Cambridge University Press, Cambridge (2013)

69. Pad array - MATLAB padarray - MathWorks Deutschland. https://de.mathworks.com/help/images/ref/padarray.html (2020). Accessed 29 November 2020

70. Kokhanovsky, A.A., Weichert, R., Heuer, M., Witt, W.: Angular spectrum of light transmitted through turbid media: theory and experiment. Applied optics (2001). https://doi.org/10.1364/AO.40.002595

71. Zhao, Y., Cao, L., Zhang, H., Kong, D., Jin, G.: Accurate calculation of computer-generated holograms using angular-spectrum layer-oriented method. Optics express (2015). https://doi.org/10.1364/OE.23.025440

72. Sung, Y., Lue, N., Hamza, B., Martel, J., Irimia, D., Dasari, R.R., Choi, W., Yaqoob, Z., So, P.: Three-Dimensional Holographic Refractive-Index Measurement of Continuously Flowing Cells in a Microfluidic Channel. Physical review applied (2014). https://doi.org/10.1103/PhysRevApplied.1.014002

73. Resize image - MATLAB imresize - MathWorks Deutschland. https://de.mathworks.com/help/images/ref/imresize.html (2020). Accessed 27 November 2020

74. Save and Load Parts of Variables in MAT-Files - MATLAB & Simulink - MathWorks Deutschland. https://de.mathworks.com/help/matlab/import_export/load-parts-of-variables-from-mat-files.html (2020). Accessed 29 November 2020

75. IEEE: 2009 International joint conference on neural networks (2009)

76. Memory information - MATLAB memory - MathWorks Deutschland. https://de.mathworks.com/help/matlab/ref/memory.html (2021). Accessed 3 January 2021

77. Born, M., Wolf, E.: Principles of optics. Electromagnetic theory of propagation, interference and diffraction of light / by Max Born and Emil Wolf with contributions by A. B. Bhatia ...[et al.], 6th edn. Cambridge University Press, Cambridge (1997, 1980)

78. Hecht, E., Zając, A.: Optics, 2nd edn. Addison-Wesley Pub. Co, Reading, Mass. (1987)

79. Shack-Hartmann Wavefront Sensors (2020). Accessed 27 November 2020

80. NewView 9000. https://www.zygo.com/products/metrology-systems/3d-optical-profilers/newview-9000 (2021). Accessed 18 January 2021

81. Create high dynamic range image - MATLAB makehdr - MathWorks Deutschland. https://de.mathworks.com/help/images/ref/makehdr.html (2020). Accessed 27 November 2020

82. R. W. Gerchberg: A practical algorithm for the determination of phase from image and diffraction plane pictures. Optik 35, 237–246 (1972)

83. Mean-squared error - MATLAB immse - MathWorks Deutschland. https://de.mathworks.com/help/images/ref/immse.html (2021). Accessed 8 January 2021

84. Zhou, Y., Feng, S., Ma, Q., Yuan, C.: Image edge enhancement using Airy spiral filter. In: Imaging and Applied Optics 2016. Imaging Systems and Applications, Heidelberg, IT1F.2. OSA, Washington, D.C. https://doi.org/10.1364/ISA.2016.IT1F.2

85. Falaggis, K., Kozacki, T., Kujawinska, M.: Computation of highly off-axis diffracted fields using the band-limited angular spectrum method with suppressed Gibbs related artifacts. Applied optics (2013). https://doi.org/10.1364/AO.52.003288

86. Kozacki, T., Falaggis, K., Kujawinska, M.: Computation of diffracted fields for the case of high numerical aperture using the angular spectrum method. Applied optics 51, 7080–7088 (2012)

87. Matsushima, K.: Shifted angular spectrum method for off-axis numerical propagation. Opt. Express 18, 18453–18463 (2010)

88. Matsushima, K., Schimmel, H., Wyrowski, F.: Fast calculation method for optical diffraction on tilted planes by use of the angular spectrum of plane waves. Journal of the Optical Society of America. A, Optics, image science, and vision (2003). https://doi.org/10.1364/JOSAA.20.001755

89. Luo, Y., Mengu, D., Yardimci, N.T., Rivenson, Y., Veli, M., Jarrahi, M., Ozcan, A.: Design of task-specific optical systems using broadband diffractive neural networks. Light, science & applications (2019). https://doi.org/10.1038/s41377-019-0223-1

9 Table of Figures

Figure 1-1	Number of publications per year for the search topics of „optical neural network“ in blue and „deep learning“ in orange. The data are taken from web of science [7].	7
Figure 1-2	Number of publications per year for the search topic of „diffractive neural networks“. The data are taken from web of science [7].	7
Figure 2-1	Generalized illustration of an artificial neural network neuron with three inputs.	10
Figure 2-2	Illustration of a shallow neural network with 4 inputs x_j, one hidden layer, and one output layer with one output y.	12
Figure 2-3	Three neurons in a network of layer $l-1$, layer l and the layer $l+1$, with common indexing used for weights, biases, activations.	12
Figure 2-4	Illustration of a shallow neural network with four inputs x_j, two hidden layers, a classification layer and three output classes.	14
Figure 2-5	Sketch of a typical classification network with three classes. The softmax activation function is abreviated by sf, any other arbitrary activiation function by f.	16
Figure 2-6	Sketch of a typical regression network, where arbitrary activation functions are indicated by f. As loss function the MSE function is used.	17
Figure 2-7	An arbitrary function $L(v)$ representing the loss of a network.	18
Figure 2-8	Illustration of the first training epoch of M samples organized into smaller batches b, each with N samples.	20
Figure 2-9	Three concatenated neurons with indices. The observation point is neuron j. Only one neuron per layer is shown.	22
Figure 2-10	Binary step activation function.	25
Figure 2-11	A linear activation function $f(z)=z$ and ist derivative $f'(z)=1$.	26
Figure 2-12	Rectifing Linear Unit (ReLU) activiation function and its derivative	27
Figure 2-13	Sigmoid activation function $\sigma(z)$ and its derivative.	27
Figure 2-14	The hyperbolic tangent (tanh) activation function and its derivative	28
Figure 2-15	Example of the solution space transformation by a layer of ReLU activation function. In a) the three inputs can not be divided by linear regression. In b) the transformed solution space at $a_{1,2}$ is shown. Now, a line can be drawn to seperate the classes A and B. In c) the network performing this transormation is shown.	29
Figure 2-16	Representation of a convolution of the kernel h with a matrix u. The result is the matrix g. Here shown are only two calculation steps for the valid region.	31

Figure 2-17 Image of a sheep a), applied sobel filter in x- and y-direction to visualize the gradient of the grayscale values in vertical direction b). 31

Figure 2-18 Architecture of LeNet-5 a convolutional neural network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.[30] 32

Figure 2-19 A convolution layer and a pooling layer in a CNN. The input a_k^{l-1} with two channels k is convolved with the three kernels k_{kj}^l to produce the six maps z_j^l. Not shown here is the addition of the bias b_j^l to each channel in z_j^l. A ReLU layer is applied tor produce x_j^l with the same channel count, which is then pooled to produce the output map a_j^l, also with the same channel count. 34

Figure 3-1 Taxonomy of PNN approaches and associated proofs of concept, indicating the hardware implementation (free space optics or integrated photonics) and the operation mode (inference only or trainable). Only the types of neural networks for which a photonic version has been demonstrated in the literature are reported. [8] 37

Figure 3-2 Broadcast-and-weight protocol and experiment of Tait et al.. (a) Concept of a broadcast-and-weight network with modulators used as neurons. MRR: microring resonator, BPD: balanced photodiode, LD: laser diode, MZM: Mach-Zehnder modulator, WDM: wavelength-division multiplexer. (b) Micrograph of 4-node recurrent broadcast-and-weight network with 16 tunable microring (MRR) weights and fiber-to-chip grating couplers. (c) Scanning electron micrograph of 1:4 splitter. (d) Experimental setup with two off-chip MZM neurons and one external input. Signals are wavelength-multiplexed in an arrayed waveguide grating (AWG) and coupled into a 2×3 subnetwork with MRR weights, w_{11}, w_{12}, etc. Neuron state is represented by voltages s_2 and s_2 across low-pass filtered transimpedance amplifiers, which receive inputs from the balanced photodetectors of each MRR weight bank. [40] 38

Figure 3-3 Illustration of Optical Interference Unit a. Optical micrograph of an experimentally fabricated 22-mode on-chip optical interference unit; the physical region where the optical neural network program exists is highlighted in grey. The system acts as an optical field-programmable gate array–a test bed for optical experiments. b. Schematic illustration of the optical neural network program demonstrated here which realizes both matrix multiplication and amplification fully optically. c. Schematic illustration of a single phase shifter in the Mach-Zehnder Interferometer (MZI) and the transmission curve for tuning the internalphase shifter of the MZI. [41] 39

Figure 3-4 Fully functioning two-layer AONN experimental setup of Zou et al. with a nonlinear element implemented as using laser-cooled atoms with electromagnetically induced transparency indicated as MOT[45]. 40

Figure 3-5 Diffractive deep neural networks (D^2NNs). a) A D^2NN comprises multiple transmissive (or reflective) layers, where each point on a given layer acts as a neuron, with a complex-valued transmission (or reflection) coefficient. The transmission or reflection coefficients of each layer can be trained by using deep learning to perform a function between the input and output planes of the network. After this learning phase, the D^2NN design is fixed; once fabricated or 3D-printed, it performs the learned function at the speed of light. b) Trained and experimentally implemented D^2NN classifier for handwritten digits and fashion products. [9] 41

Figure 3-6 Salient object detection with Fourier-space diffractive deep neural network (F-D^2NN). The F-D^2NN optical image processing module is formed by inserting the D^2NN along with a photorefractive crystal (SBN:60) at the Fourier plane of an optical system under coherent light.
F-D^2NNs can achieve all optical segmentation of the salient objects for the target scene after deep learning design of modulation layers. [47] 42

Figure 3-7 Three neurons in a diffractive deep neutral network with the indexing used in this work. 43

Figure 3-8 Coordinate system for two physically space diffractive neural network layers with the radial distance of a neuron k to a neuron j indicated by r. 43

Figure 3-9 Forward Propagation of a D^2NN with three difractive layers according to [9] and [48]. An incoming monochromatic plane wave is modulated by $\boldsymbol{t}^0$ in amplitude to form the input image. Through several layers of propagation $\boldsymbol{w}^l$, modulation $\boldsymbol{t}^l$ and activation $\boldsymbol{f}^l$ the diffracted optical wave $\boldsymbol{a}^{L+1}$ falls onto an observation screen. 44

Figure 3-10 Classification Layer of a D^2NN at the layer $L+1$ of the network with each image point resampling an output neuron y_j. Every output neuron that falls into the region c_n is assigned to the class number n. 45

Figure 3-11 Basic binary optical grating, where the grating has either a changing amplitude from 1 to 0 or a changing phase from 0 to π. 47

Figure 3-12 Sketch of two discrete spatial frequencies resulting from a sampled surface. 48

Figure 3-13 Multiple diffraction images due to surface discretization with crosstalk in the overlap region a) and with no crosstalk in b). 48

Figure 3-14 Illustration of a convolutional unit in an F-D^2NN; a) shows the optical setup with two lenses at a modulating kernel in the focal (Fourier) plane; b) example of an image modulated in the Fourier plane with a high-pass filter. 52

Figure 3-15 Minimum sampling distance Δs for the Rayleigh-Sommerfeld integral for several distances Δz as a function of the aperture size a shown as continous line and the corresponding number for weight 54

multiplications or calculations per diffraction layer is shown as dashed lines.

Figure 3-16 Concept of a multi-kernel convolution layer unit based on a diffraction grating, a lens, multiple diffractive kernels, and a lens array. 55

Figure 4-1 Imaging of a point source through an aperture, where a) only takes geometrical ray optics into account and b) shows an intensity pattern according to wave optics. 57

Figure 4-2 Monochromatic wave at a fixed point represented as a) a harmonic wave with frequency ν and b) as complex amplitude phasor with the vector length A and phase φ. 60

Figure 4-3 Solutions to the Helmholtz equation; a) a plane wave travelling in z-direction with the spatial periodicity λ and b) a spherical wave at the origin z_0, x_0 and $k_x = k_z$. 61

Figure 4-4 Integration Surface S surrounding the integration volume V, around an optical disturbance center P_0. Where $\vec{n}$ are the normal vectors on each surface, S_ε is the surface with radius ε around P_0 to resolve the singularity problem of the Green's function.P_1 is the pointe of the disturbance at an arbitrary point. 63

Figure 4-5 Screen with aperture Σ, where the surface S_1 lies directly behind the aperture and has the normal vector $\vec{n}$, the surface S_2 is a spherical section around the observation point P_0 and the radius R. Point P_1 lies on S_1 and inside Σ and P_2 lies left of P_1. 64

Figure 4-6 Example for the application of the Fresnel-Kirchhoff integral, where a plane wave illuminates the Aperture Σ. The diffracted wave is observed at P_0 with respect to all illuminated points P_1 in Σ. 65

Figure 4-7 Visualization of the diffraction problem with the Green's function $G_\pm$ used by Sommerfeld. 67

Figure 4-8 Block symbol of system with one input $u(x,y)$ and output $g(x,y)$ 69

Figure 4-9 Discretization of a one-dimensional continuous signal u with shifted δ-functions by Δx and weighted with the local amplitude at $n\Delta x$. 73

Figure 4-10 a) Amplitude output of a one-dimensional DFT/FFT operation with the alias centered at the sampling frequency and the Nyquist Frequency $1/2\Delta_x$. In b) the corrected spectrum with the lower frequency half of the alias as negative frequency range ν^-. The continuous spectrum of a cont. Fourier transform is indicated as a solid line and the sampled spectrum as dots representing the power in each frequency bin of width ν_x and c) shows the zero-order shifting for a two-dimensional FFT spectra in three steps. 75

Figure 4-11 Wraparound contributions at P_0 due to no zero-padding. If zero-padding is added to the computation window the contribution indicated by the dotted arrows falls onto the screen and not onto P_0. 76

Figure 4-12 Phase shift of a free-space transfer function $H(\nu_x, \nu_y; z)$ at $z = 100\,\lambda$ for a wavelength of $\lambda = 1$ 77

Figure 4-13 Geometrical description of an aperture field propagated to a screen with a) insufficient padding so that power of the replicas influence the output field in the computation window and b) a sufficient padding so that no spectral components influence the field at the observation plane in a specific observation window. 78

Figure 4-14 Phase of FSP functions at $z = 500\,\lambda$ and $\lambda = 1$ with a) oversampled at $\Delta\nu_{x,y} = 50 \cdot 10^{-3}\,\frac{1}{\lambda}$ $(2000^2\,px)$ and b) undersampled at $\Delta\nu_{x,y} = 8.3 \cdot 10^{-3}\frac{1}{\lambda}$ $(120^2\,px)$, where the zero-frequency is in the bottom-left corner 79

Figure 4-15 Minimal FSP function sampling requirement (4-73) plotted for different observation plane distances z, arbitrary small aperture size and in units of wavelengths. 81

Figure 4-16 Phase amplitude of two FSP functions at a) $1200\,\lambda$ an b) at $3600\,\lambda$. The red ellipse indicates the valid region of ν_x calculated by (4-78). The blue ellipse indicates the valid region for ν_y calculated by (4-79). ν_m is the maximum spatial frequency $\nu_{x,max} = \nu_{y,max}$ for the case of symmetric sampling of the calculation window. 82

Figure 4-17 Aperture & observation plane with two points with the greatest distance i.e., with the maximal needed local frequency f_{ν_x} and f_{ν_y} 84

Figure 4-18 Types of wavefront modulations. a) Phase modulation $\Delta\varphi$ by local variations Δd of the travel distance inside a medium of constant refractive index n_1. b) Phase modulation $\Delta\varphi$ by local variations of the refractive index Δn of a medium with constant thickness. 86

Figure 4-19 Two-dimensional sketch of a plano-convex lens in thin lens approximation with the refractive index n_1, aperture diameter D, apex thickness d_0 and focal length f'. The convex surface has the radius R. 87

Figure 4-20 Modulation function of an ideal lens, with a diameter D of 100 mm, refractive index of $n_1 = 1.4$ and focal length $f' = 10^5\,mm$ for a wavelength of $\lambda = 632.8\,nm$. a) shows the phase angle of the modulation function and b) the amplitude truncated by the $circ\left(\frac{D}{2}\right)\,function$. 88

Figure 4-21 Illustration of the subscript convention of a padded calculation window u_P. Where u is the actual window, surrounded by P_x invalid data points in x-direction and P_y invalid data points in y-direction. The total size of u_P is $N_P \times M_P$. 90

Figure 4-22 Flowchart of the BLAS/AS propagation algorithm. 91

Figure 4-23 Flowchart of the algorithm for creating the FSP transfer function H. 93

Figure 4-24	Principle of the satandalone wave propagation algorithm, where a field distribution gets repeatedly propagated and modulated. The field is evaluated at the output plane.	94
Figure 4-25	Flowchart of the standalone BLAS simulation algorithm.	97
Figure 5-1	Calculation speed measurement of three diffraction calculation methods. The methods are the rigorous calculation of the Rayleigh-Sommerfeld integral (RS), the convolution-based direct integration method (DI) and the band limited angular spectrum method (BLAS).	102
Figure 5-2	Composition of computation steps of the BLAS method as stacked bars. The percentages at each element denote the contribution of each calcuation to the overall calculation time.	103
Figure 5-3	Isolated calculation times for the computation of the optical field propagation of the BLAS method. The functions used are one FFT, a element-wise matrix multiplication and a IFFT. The calculation time is measured for several Matrix samplings. The total number of matrix elemnts is the square of the sampling points.	104
Figure 5-4	Measurement of the RAM usage by increasing sizes of single and double precision complex matrices. Whereby the horizontal axis is the number of points in one direction, the actual matrix size would be the value squared.	104
Figure 5-5	Calculated sampling conditions for the RS (dashed) and BLAS (continous) method in wavelengths $[\lambda]$ for a unitary aperture size of 1.	105
Figure 5-6	Forward Propagation of a D^2NN with three diffractive layers according to [9] and [48]. An incoming monochromatic plane wave is modulated by $\boldsymbol{t}^0$ in amplitude to form the input image. Through several layers of propagation $\boldsymbol{w}^l$, modulation $\boldsymbol{t}^l$ and activation $\boldsymbol{f}^l$ the diffracted optical wave $\boldsymbol{a}^{L+1}$ falls onto an observation screen.	107
Figure 5-7	Setup for measuring the focal length of a modelled lens aperture illuminated by a monochromatic plane wave. The observation plane is in the x-z-plane and chosen so that the focal point ideally is in the observation plane center. The optical axis is indicated by o.a. and the direction of the incoming wavefront by k.	111
Figure 5-8	Setup for observing an airy pattern created by a plane wave passing a circular aperture and a modelled lens. The observation plane is located at the focal plane of the lens.	111
Figure 5-9	Cross-section along the x-z-plane of a simulated monochromatic focused wavefront by a modelled lens with a focal length of $f' = 200\,mm$. The aperture diameter is $D = 8\,mm$ and the wavelength is $\lambda = 633\,nm$. The shown intensity is in a logarithmic scale of normalized intensity.	112
Figure 5-10	Intensity in the x-y-plane of a simulated focal spot in the focal plane. The aperture size is $D = 1\,mm$, the focal length of the lens is $f' =$	113

$20\,mm$, the wavelength is $\lambda = 633\,nm$ and the spatial sampling is $\Delta s = \lambda/2$.

Figure 5-11 Intensity profile of a simulated focal spot in the focal plane. The aperture size is $D = 1\,mm$, the focal length of the lens is $f' = 20\,mm$, the wavelength is $\lambda = 633\,nm$ and the spatial sampling is $\Delta s = \lambda/2$.The position of the first intensity minimum to the right of the cnter is indicated by a dashed line. 113

Figure 5-12 Intensity in the x-y-plane of a simulated focal spot in the focal plane. The intensity is shown in a logarithmic scale. The aperture size is $D = 1\,mm$, the focal length of the lens is $f' = 20\,mm$, the wavelength is $\lambda = 633\,nm$ and the spatail sampling is $\Delta s = \lambda/2$. 114

Figure 5-13 Setup for observing the diffractive pattern of a grating under illumination of a plane wave, that is additionally focused by a lens with the focal length f'. 115

Figure 5-14 Intensity cross-section in the x-z-plane of the diffraction pattern caused by a binary amplitude grating and a focusing lens. Diffraction orders up to second orders are visible. 116

Figure 5-15 Intensity cross-section in the x-y-plane of the diffraction pattern caused by a binary amplitude grating and a focusing lens. Diffraction up to second orders are visible. 116

Figure 5-16 Intensity profile of the diffraction pattern caused by a binary amplitude grating and a focusing lens. Diffraction up to second orders are visible. The dashed lines indicate the calculated first order diffraction maxima. 116

Figure 5-17 Intensity cross-section in the x-y-plane of the diffraction pattern caused by a binary phase grating and a focusing lens. Only the $\pm$ first orders are visible, the 0^{th}-order is suppressed. 117

Figure 5-18 Intensity profile of the diffraction pattern caused by a binary phase amplitude grating and a focusing lens in a logarithmic scale. Diffraction up to second orders are visible. The dashed lines indicate the calculated first order diffraction maxima. 117

Figure 5-19 Experimental setup for a symmetric 4-f system with two lenses. The input is a plane monochromatic wave. The wave is first observed in the Fourier plane and directly behind the second lens aperture and the in the observation plane 2 at a distance f' form the last aperture. 118

Figure 5-20 1951 USAF resolution test chart used in this experiment with additional zero-padding. 121

Figure 5-21 Simulation of a symmetric 4-f imaging system with the focal lengths $f' = 50\,mm$, a lens with the diameter of $20\,mm$ at a sampling distance of $\Delta s = 10\lambda$ and a simulation wavelength of $\lambda = 633\,nm$. In a) the intensity image in the Fourier plane is shown and in b) the corresponding intensity image at the observation plane is shown. In 122

c) the spectral image is truncated by an aperture with the radius of $r_{AP,2} = 2.5\,mm$ and d) shows the corresponding image.

Figure 5-22 Fourier space alias locations. Whereby a) shows the simulation result of Figure 5-21 b) with numbered squares that roughly represent each image segment and b) shows the input image with the corresponding image segments correlated by numbers in both images. 122

Figure 5-23 Simulation of a symmetric 4-f imaging system with the focal lengths $f' = 50\,mm$, a lens with the diameter of $20\,mm$ at a sampling distance of $\Delta s = 5\lambda$ and a simulation wavelength of $\lambda = 633\,nm$. In a) the intensity image in the Fourier plane is shown and in b) the corresponding intensity image at the observation plane is shown. In c) the spectral image is truncated by an aperture with the radius of $r_{AP,2} = 5\,mm$ and d) shows the corresponding image. 123

Figure 5-24 Simulation of a symmetric 4-f imaging system with the focal lengths $f' = 50\,mm$, a lens with a diameter of $20\,mm$ at sampling distance of $\Delta s = 2.5\lambda$ and a simulation wavelength of $\lambda = 633\,nm$. In a) the intensity image in the Fourier plane is shown and in b) the corresponding intensity image at the observation plane is shown. 124

Figure 5-25 Intensity images of Fourier transformations of the USAF chart using the FFT in a), c), d) and a simulated lens and free-space propagation in b), d), f). The FFT images are scaled to show the same frequency band as the simulated images and are shown in units of $[mm^{-1}]$. The focal length of the lens in a) is $f_1' = 25\,mm$, in c) the focal length is $f_2' = 50\,mm$, in e) the focal length is $f_3' = 100\,mm$. 125

Figure 5-26 Simulated image of a USAF chart 100 mm behind a 4-f system with the focal lengths of $f' = 100\,mm$. The sampling is $\Delta s = 2.5\,\lambda$ and a window size of $20 \times 20\,mm^2$. 126

Figure 5-27 Quarter sections of lens apertures with a focal length of $f' = 100\,mm$. In a) a lens is shown which is sampled at $\Delta s = 3.2\,\mu m$ and in b) the sampling distance is $\Delta s = 63.4\,\mu m$. 127

Figure 5-28 Conceptual design of a multi-kernel convolutional unit for a *D²NN*. 130

Figure 5-29 Images of the modulating planes in the third experimental setup. The circular aperture function with the diameter of $2\,mm$ is shown in a). A magnified image of the binary phase grating is shown in b). The phase of the focusing with $f' = 100\,mm$ lens is shown in c). The lens array with a lens displacement of $2.5\,mm$ is shown in d). 132

Figure 5-30 Scaled USAF chart the testing the simulation of a multi-kernel convolution unit. 133

Figure 5-31 Cross section of a three dimensional BLAS simulation of a conceptual convolution unit. The input is a plane wave truncated by a $2\,mm$ aperture which is modulated by a binary phase grating. A lens at a distance of $100\,mm$ and with a focal length of $f' = 100\,mm$ focuses 133

each image in the Fourier plane. A lens array located $2 \cdot f'$ from the aperture collimates each image separatly.

Figure 5-32 Magnified image of the Fourier plane in Figure 5-31. The theoretical focal plane is indicated as solid white line at $z = 200.02\,mm$ and the measured focal as dashed line at $z = 202\,mm$. 134

Figure 5-33 Intensity image of a USAF imaged by a convolutional unit, with the focal lengths of $f' = 100\,mm$, a spatial sampling of $\Delta s = 5\lambda$, a window size of $10 \times 10\,mm^2$ and a binary phase diffraction grating. 135

Figure 5-34 Phase image of a USAF imaged by a convolutional unit, with the focal lengths of $f' = 100\,mm$, a spatial sampling of $\Delta s = 5\lambda$, a window size of $10 \times 10\,mm^2$ and a binary phase diffraction grating. 135

Figure 5-35 Experimental setup with a laser at $\lambda = 632.8\,nm$ is collimated and widened with two lenses f_1, f_2 and an aperture stop A_1. A second aperture stop A_2 truncates the beam for a homogenous intensity profil. With a beam splitter BS one path of the laser beam is measured with a wavefront sensor WFS and the other passes a grating and the interference pattern is observed at a diffusing screen. 137

Figure 5-36 Surface topology of a chromium grating measured by a WLI. a) show the complet data of $6 \times 6\,mm^2$ and b) a subsection $0.9 \times 0.9\,mm^2$ of a) to visualize the individual grating lines. 139

Figure 5-37 Amplitude modulation of the aperture function, derived from the measurements of Figure 5-36. The total field size is $1.0418 \times 1.0418\,mm^2$. 140

Figure 5-38 a) WFS measurement of the illumination beam phase and b) the bicubic interpolation of the data. 140

Figure 5-39 a)WFS measurement of the illumination beam intensity distribution in arbitrary units and b) the bicubic interpolation of the data. 141

Figure 5-40 HDR image of the laser beam without diffraction grating the the observation screen at distance $z = 30\,mm$. 141

Figure 5-41 HDR image of the diffracted laser beam by the chromium grating at the observation screen $z = 30\,mm$. 142

Figure 5-42 Simulation results of a laser beam diffracted by a grating and observed at a screen at a distance of $z = 30\,mm$ 142

Figure 5-43 Cross section of measured and simulated diffraction pattern. The solid line indicates the calculated first order diffraction distance of $\pm 2.54\,mm$ at the screen. 143

Figure 5-44 Generalized Gerchberg-Saxton phase retrieval algorithm based on [82]. Where A,B,C and D are the variables used in the software implementation. The letters P and I denote the phase and intensity of the complex field, respectivly. 144

Figure 5-45 GS algorithm gaussian source intensity in a) and the target image is shown in b). 146

Figure 5-46 Hologram of Figure 5-45 approximated by the GS algorithm in a). Hologram multiplied with a lens modulation function for recunstruction shown in b). A magnified version of b) with a limiting aperture is shown in c), whereby the aperture radius is defined by the radius of the source intensity at which the intensity value is 10^{-3}. 146

Figure 5-47 Image of the first CGH extracted from the GS algorithm, using a Fresnel approximation. In a) the image in a linear scale is shown and in b) the same image in a logarithmic scale to visualize the noise. 147

Figure 5-48 Hologram of the USAF chart in Figure 5-20 approximated by the GS algorithm in a). Hologram multiplied with a lens modulation function for reconstruction shown in b). A magnified version of b) with a limiting aperture is shown in c), whereby the aperture radius is defined by the radius of the source intensity at which the intensity value is 10^{-3}. 148

Figure 5-49 Image of the USAF chart CGH extracted from the GS algorithm, using a Fresnel approximation. In a) the image in a linear scale is shown and in b) the same image in a logarithmic scale to visualize the noise. 148

Figure 5-50 Intensity image of a GS CGH of Figure 5-47 using a lens as modulation and the BLAS method for calculation. 149

Figure 5-51 Intensity image in a logarithmic scale of a GS CGH of Figure 5-47 using a lens as modulation and the BLAS method for calculation. 150

Figure 5-52 Intensity image of a GS CGH of Figure 5-48 using a lens as modulation and the BLAS method for calculation. 151

Figure 5-53 Intensity image in a logarithmic scale of a GS CGH of Figure 5-48 using a lens as modulation and the BLAS method for calculation. 151