A machine learning approach for transient imaging reconstruction

Miguel Ángel Cosculluela Gracia

Trabajo de fin de Máster en Modelización e Investigación Matemática, Estadística y Computación
Universidad de Zaragoza

Directores del trabajo: Dr. Julio Marco Murria y Prof. Dr. Diego Gutiérrez Pérez

9 de febrero de 2021

Abstract

The recent advances in non-line-of-sight imaging have made it possible to reconstruct scenes hidden around a corner, with potential applications in e.g. autonomous driving or medical imaging. By operating at frame rates comparable to the speed of light, recent virtual-wave propagation methods leverage the temporal footprint of indirect light transport at a visible auxiliary surface to take virtual photos of objects hidden from the observer. Despite these advances, these methods have a critical computational bottleneck: The reconstruction quality and the computational performance are highly dependent on the resolution of the capture grid, which is typically discretized in space and time, leading to high processing and memory constraints.

Inspired by recent machine learning techniques, in this work we propose a new computational imaging method to address these limitations. For this purpose we propose to learn implicit representations of the captured data using neural networks, allowing us to convert the discrete space of the captured data into a continuous one. However, working directly with the captured data is a complex task due to its huge size and its high dynamic range values. In order to avoid these problems, we leverage recent wave-based phasor-field imaging methods to transform the time-resolved captured data into sets of 2D complex-valued fields (i.e. phasor fields) at different frequencies, which provides a more favorable representation for machine learning methods.

Under our implicit representation formulation, we analyze the performance of different neural network models to represent the complex structure of phasor fields, starting from simpler representations, and iteratively providing more powerful models to add support for the complexity of the data. We demonstrate how recent machine learning techniques based on multilayer perceptrons with sine activation functions are capable of representing phasor fields analytically in both spatial and temporal frequency domains, and integrate them into the phasor-field framework to reconstruct hidden geometry. We finally test this neural model in different scenes, and measure its performance at higher resolutions not seen by the captured data. We show how the model is able to analytically upsample all dimensions, and demonstrate how our implicit representation additionally works as a denoiser of the source discretized phasor field.

Resumen

Los recientes avances en imagen non-line-of-sight han hecho posible la reconstrucción de escenas ocultas a través de una esquina, con la potencial aplicación en conducción autónoma o imagen medica. Al operar con fotogramas por segundo cercanos a la velocidad de la luz, recientes métodos de propagación virtual de ondas aprovechan la huella temporal del transporte de luz en una superficie auxiliar para tomar fotones virtuales de objetos ocultos al observador. A pesar de estos avances, estos métodos tienen un cuello de botella critico: La calidad reconstrucción y el coste de computo son altamente dependientes de la resolución de la malla de captura, la cual suele esta discretizada en espacio y tiempo, lo que conlleva grandes limitaciones de procesamiento y memoria.

Inspirados en las recientes técnicas de aprendizaje automático, en este trabajo proponemos un nuevo método de imagen computacional para hacer frente a estas limitaciones. Para este propósito proponemos aprender representaciones implícitas de los datos capturados usando redes neuronales, permitiéndonos convertir el espacio discreto de los datos capturados en un espacio continuo. Sin embargo, trabajar directamente con los datos capturados es una tarea compleja debido a su gran tamaño y al alto rango dinámico de sus valores. Para evitar estos problemas aprovechamos el reciente método de imagen de los campos de fasores basados en ondas para transformar los datos capturados resueltos en tiempo en un conjunto de campos 2D de valores complejos (como son los campos de fasores) a diferentes frecuencias, lo cual provee una representación más favorable para los métodos de aprendizaje automático.

Siguiendo nuestra formulación de representación implícita, hemos analizado el rendimiento de diferentes modelos de redes neuronales para representar la compleja estructura de los campos de fasores, empezando por representaciones simples, y proporcionando de forma iterativa modelos más potentes para añadir soporte para la complejidad de los datos. Demostramos como las técnicas recientes de aprendizaje automático basadas en preceptrones multicapa con funciones de activación sinusoidales son capaces de representar un campo de fasores analíticamente en los dominios espaciales y temporales, e integrarlas dentro del marco de los campos de fasores para reconstruir geometría oculta. Finalmente probamos este modelo de red neuronal con diferentes escenas y medimos su desempeño con mayores resoluciones que no han sido usadas en el entrenamiento. Mostramos como el modelo es capaz de generar más muestras en todas las dimensiones y demostramos como nuestra representación implícita además funciona como un método para eliminar ruido del campo de fasores discretizado.

To my family and friends in the laboratory that helped me during the realization of this master's thesis. I would like to thank my directors, who orientated me, and my partner who supported me during this time and who has given me the strength to keep on.

Contents

Chapter 1

Introduction

Transient imaging methods [1] are focused on analyzing and capturing time resolved profiles of the propagation of ultrashort light pulses emitted by ultra-fast lasers at temporal resolutions close to the speed of light. These profiles consist in impulse response functions of a scene in time. The information that can be obtained from them has several applications such as depth estimation, vision through turbid media or reconstruction of hidden objects. Concretely, non-line-of-sight (NLOS) reconstruction methods capture the impulse response function of a hidden scene from a visible diffuse wall (known as relay wall, in the figure 1.1 it can be seen an example of the capture setup for a hidden scene) and, by analyzing it, they are able to recover information of the hidden scene such as the position, the geometry or the material properties. NLOS methods have multiple potential applications such as lunar cave exploration, rescue operations in disaster situations or autonomous driving.

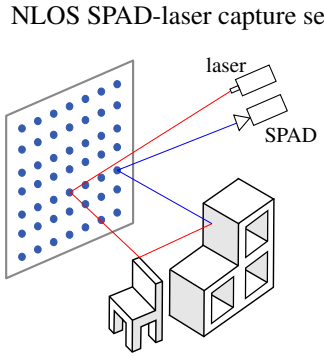

Figure 1.1: Sample setup for capturing a NLOS scene. An ultrashort light pulse is emitted to a relay wall (square with dots) and the sensor captures the light resolved in time reflected by the scene to the relay wall. Image adapted from Liu et al.[2].

NLOS methods analyze the light paths from the relay wall that come from a hidden scene to estimate the position of geometry or other properties of the hidden scene such as the type of material. The recent work of phasor field by Liu et al. [2] presented a wave-based framework that overcome all previous methods and is successfully able to work with complex scenes where multiple inter-reflections occur. The framework transforms the temporal domain of the impulse response function into a wave-optics domain converting a NLOS scene into a classical line-of-sight (LOS) scene as if it was observed from the relay wall perspective, enable the use of well-established LOS optics methods. Furthermore, the framework allows virtualizing different light models that are virtually propagated to the scene and, through virtual cameras and lens models, is able to focus and reconstruct the hidden scene. Despite these advances, the

NLOS field still has some problems as the limitation of the capture resolution which is, nowadays, one of the biggest bottlenecks to increase the quality and details in the reconstructions.

The capture process of the impulse response function in NLOS scenes is done by scanning a relay wall sequentially with a laser and a sensor, moving them independently point by point. The sequential process is a bottleneck because it is necessary to move the sensor and calibrate it with the laser for each of the points. This makes the capture process slow and unpractical for capturing high density impulse response function. This limitation of the spatial resolution of the impulse response function also affects to the quality of the reconstructions of the NLOS methods because the capture resolution limits the maximum achievable resolution in the reconstruction.

Alleviating the bottleneck in the capture process would allow the NLOS field to be used with even more complex and detailed scenes which previously could not be reconstructed due to the low capture resolution, allowing to improve the quality of NLOS reconstructions. This problem can be addressed from two sides, modifying the current capture hardware or by developing novel computational imaging methods. In the first option, in order to modify or create new hardware it is required to build prototypes and design prototypes that are often too expensive or impractical to manufacture. Moreover, the new hardware will must be extensively tested capturing real data what limits its development to the places with capture hardware. The second option is based on designing new computational methods that operate under the existing hardware constraints, to take advantage of the characteristics of light transport information in order to improve performance in the imaging process.

Recovering a high dense signal or function from its low sample version is a known problem in multiple fields where the original function is measured with different methods which are not able to sample enough points to recover the original shape of the function. A recent example of this problem is the reconstruction of the black hole image [3] obtained from a highly sparse amount of points captured from radio-telescopes across the entire world. The recovering process is commonly done with machine learning methods due to the complexity of the functions to recover and the flexibility to adapt and learn the peculiarities of the original data.

Learning methods have shown along the years its potential in fields like computer vision, graphics or imaging, being able to solve problems such as superresolution [4], image segmentation [5], denoising [6] or depth estimation [7] with high frequency details like borders in images or geometry or strong illumination changes in scenes. One of the techniques that are able to learn this type of non-linearities are the neural networks. They have demonstrated their ability of filling gaps between samples. This is a good option for our problem where we want to increase the samples of a discrete phasor field obtaining a more density version. An example is that, in the case of superresolution, a neural network increases the resolution of an image hallucinating new pixels [8]. Other option for increasing the resolution is the idea of being able, through a neural network, to sample new points from the original learned data. This has been tested in different domains such as textures sampling [9], 3D volumetric spaces sampling [10, 11] or implicit models learning of different types of data as images, signals or 3D volumes [12]. However, all these methods are designed for other problems, working in real domain or with standard images, making that none of them can be directly used in our case. However, in this work we will follow these ideas and formulate this problem as a transformation of a discrete impulse response function into a continuous function learning an implicit representation.

However, applying this directly to the impulse response function is a difficult problem. The response function has the disadvantage of its size and its high dynamic range united to the fact that it is not necessary to sample the temporal dimension to higher resolutions (see section 4.2 for further details). To avoid this problem we take advantage of the first phase of the phasor field framework that transforms the impulse response function by integrating it with a virtual light profile into a phasor field for a concrete frequency in the relay wall. With this change we alleviate the previous problems working with low range values in complex space and without a long dimension like the temporal one. With this transformation we can now learn an implicit representation for the spatial and frequency domain allowing us to upsample the phasor field in all this dimensions.

To summarize, in this work we present a method to solve a current problem in the NLOS field: the limitation of capturing high density impulse response function in the spatial domain. For this work we choose to work with synthetic data instead of real one due to the control and possibilities that simulation gives us. To do the simulation we use a public transient renderer [13] that virtualizes an NLOS capture system. Once the data is simulated, we begin transforming the impulse response function through the phasor field framework into a phasor with discrete resolution. To address the chosen problem, we propose to represent discrete phasor fields using implicit representations through neural network models. This representation encodes the whole phasor field in a continuous space of spatial locations and temporal frequencies, allowing us to evaluate unknown points on the phasor field that are hallucinated by our neural model. Concretely in this work we present the following contributions:

- We introduce a formulation for implicit representations of complex-valued phasor fields for NLOS reconstruction methods.
- We study different neural models behavior in representing phasor fields within this formulation, and their suitability to recover the underlying structure of light transport across both spatial and temporal frequency domains.
- We showcase the performance of neural networks with sinusoidal activations in successfully representing phasor fields in a wide variety of NLOS scenarios, and integrate our model within the phasor-field reconstruction pipeline to provide reconstructions of hidden geometry.

1.1 Thesis Background

This thesis has been done inside the research group *Graphics and Imaging Lab* within Departamento de Informática e Ingeniería de Sistemas (DIIS) at Escuela de Ingenería y Arquitectura de la Universidad de Zaragoza (EINA). The project has been supervised by Dr. Julio Marco and Prof. Dr. Diego Gutierrez.

Chapter 2

Related Work

2.1 Non-line-of-sight (NLOS) transient imaging

Non-line-of-sight (NLOS) imaging methods aim to recover properties of a scene occluded from the sensor. Different ways to recover information exist and, in this work, we will focus in transient imaging methods. A transient profile is the impulse response function of a scene resulting from illuminating the scene with an ultrashort light pulse captured by a high speed sensor. The use of transient imaging for NLOS was proposed by Kirmani et al. [14] and empirically demonstrated by Velten et al. [15] who built a whole system to capture the impulse response function and reconstruct the occluded scene with computational methods.

From this work, the field has evolved optimizing reconstruction methods such us backprojection [16] or implementing new algorithms [17, 18]. However, the types of scenes that can be reconstructed by these methods are limited, the surfaces are mostly plane and isolated to avoid problems such us interreflections between objects or walls. The work of Xin et al. [19] presented a new method that can reconstruct more complex scenes with multiple curves material, removing part of the previous problems. Other example is the work of Tsai et al. [20] where, through optimization, a triangle mesh can be transformed into an approximation of the hidden geometry without any shape restriction. Finally, the works by Liu et al. [21] and Lindell et al. [22] presented new wave-based methods which alleviate the restrictions in shape geometry, number of objects and types of material providing methods that can work with more complex. Specifically, the work by Liu et al. [21] presents a new method that allows to virtualize different types of lights and cameras by transforming the NLOS scene into a line-of-sight scene as if it was observed from the relay wall perspective. This change of paradigm allows the authors to use the vast knowledge of line-of-sight imaging, wave-optics and traditional photography in NLOS scenes.

Despite all these advances, the quality of the reconstructions is still restricted by spatial resolution of the captures. To increase the capture resolution a recent work by Renna et al. [23] presented a new 1D array sensor and a 2D array version is now in development. In this work we presented a method to improve the quality of the reconstructions without the need of capturing big impulse response function matrix saving capture time.

2.2 Transient rendering

In NLOS one of the most valuable tools that help to test and develop new methods is the simulation of the capture system and NLOS scenes and, in our case, the possibility to generate data for machine learning algorithms. The simulation is done with a rendering process. Concretely, the rendering process simulates the light and its interactions with the matter in a virtual scene and capture the light with virtual sensors. The standard rendering process used in most of the applications such as cinema or video games generates images in 2D with the light in the scene integrated. This type of render is known as steady-state rendering. Other type is the transient rendering methods which allow simulating how the light moves through a scene, rendering videos at frame rates comparable to the speed of light. However, adding the temporal dimension in transient rendering is not always possible. This is because some applications and methods developed for steady-state rendering cannot be applied easily or, in some cases, at all. To implement a transient renderer some works presented new time-resolved light transport equations [24, 25]. Other works generalize Monte-Carlo methods for steady-state rendering as path tracing to a time-resolved version [26, 13]. Other algorithms as photon mapping were also generalized to time-resolved versions [27].

The use and research in transient rendering have grown with the development of NLOS transient imaging field. Its use is extremely useful in the NLOS field due to it helps to develop and test NLOS methods. This is due to the possibility of generating transient profile in controlled conditions without the restrictions of real capture hardware and use them as reference data. Following this last idea it is possible to generate datasets and ground truth for deep learning methods. For example the work by Marco et al. [28] uses a transient renderer to generate a time-of-flight (TOF) dataset and train a deep learning model to reduce the error in TOF cameras. Other interesting work by Liang et al. [29] presented a compression method of transient profiles with a deep learning model trained with synthetic transient data. Also, the work by Galindo et al. [30] presented a public dataset with NLOS scenes. In contrast, some NLOS methods do not use the complete light paths but only the first bounce [31, 32, 33]. This type of rendering is faster and the work by Chopite et al. [34] uses this type of rendering to generate a bigger dataset and train a deep learning model to reconstruct NLOS scenes from their transient profiles.

2.3 Upsampling methods

A capture process by definition cannot obtain the real shape of the space but only get a certain number of samples (which is known as the resolution). Having a higher resolution gives more accuracy of the measured space. However, the resolution is typically limited by the capture hardware or physical restrictions in the capture process. When the resolution is large enough, the original shape can be obtained with a simple linear interpolation because the changes between samples practically follow a straight line. But in most cases the resolution is too low and obtaining the original shape is a hard problem. With the evolution of neural networks several works have demonstrated their capabilities of understanding the structure of the data and being able to upsample them. For this propose, there are two main approximations. The first approach consists in upsampling to a higher resolution for a fixed size or scale rate (e.g. duplicating resolution). The second one uses the neural network to, for a concrete input like spatial coordinates, time or frequency, get the value of the original data or, if the input is previously unknown, hallucinate new samples.

The first approach is commonly used in image super-resolution. With the idea of upsampling for fixed sizes, several neural networks models have been developed from models which only need one low resolution image as an input [8] to models which leverage the information from multiple images such as video frames [35]. If the upsampling is done over the temporal domain instead of the spatial one the effect is super-slow motion videos [36].

The second approach is more flexible than the first one because it is possible to sample in concrete zones or at any desired resolution. However, they have the cons that their models are not general and, for each scene, a new training is required. Training and sampling a model have two main ways to be implemented. In the first way the model has as an input the scene or data (or a transformation of them) and the coordinates to be sampled. This case have been tested for compression models of textures images, obtaining a latent space of them. Then the authors can decompress the latent space by sampling with neural networks [37, 9] for coordinates in the original textures or for new ones. The other way fits the scene directly in the neural network. This is known as implicit representation and has the advantage that, once the neural network is trained, the training data is no more used. This type of networks has as an input the different types of coordinates (spatial, temporal, angular, etc.) and, as the output, the corresponding values for that coordinates. Like the previous methods, this type of models can be used to generate samples in previous unknown positions. A useful case is learning 3D scenes [38] which, in render applications, avoids the requirement of rendering new views. Closer to our approach there are recent similar works which simplify the training process and improve the results [10, 11] presenting a new method to transform the coordinates and improving significantly the results. Furthermore, the work of Sitzmann et al. [12] shows the possibilities of using periodical functions instead of the classical activation one for learning high frequency details in several implicit representation problems such as image or video representation.

Chapter 3

Background

This chapter describes the main mathematical, physical, and computational aspects that this work builds upon. Firstly we introduce a basic knowledge on neural network explaining them and the model used in this work with the most common optimization method used with neural networks. Secondly we introduce the non-line-of-sight problem and explain the theory of phasor fields, a state-of-the-art method for non-line-of-sight reconstruction.

3.1 Neural network optimization

Neural networks are learning systems inspired in human neurons and their interconnections. They are able to learn a transformation and understand the relationship between the input and output data used in the training process. The capacity of learning the transformation lies on the non-linearities (small changes in the input lead to big changes in the output) that networks are able to find and learn from a dataset. One of the main capabilities of the neural networks is the generalization, since once they are trained they can be used with new unknown data with a really low computational cost. An example of this are the classification networks that can be used even in low powerful smartphones with their camera. However, this powerful method has a strong requirement for the majority of applications as neural networks require datasets with hundreds of thousands of examples.

A neural network is formed by a set of functions called neurons which are interconnected with each other. Each neuron has multiple inputs with which operates, propagating the result to the next neurons. This process is repeated from the first neurons that compute the input to the last ones which generate the output. This sequential execution allows ordering the neurons in layers. There are basically three types of layers: the input layer, the output layer and the hidden ones which are all the layers in between the input and the output. The number of these hidden layers is variable and can be finetuned depending on the problem which the neural network is trying to solve. The same strategy can be followed regarding the number of neurons. Both are hyper-parameters that need to be obtained experimentally.

In the last years, the development of neural networks has derived in a specialization of architectures and different types of networks have shown better achievements for certain types of data or problems. For example, some architectures have shown high capacity generating a compressed version of the input and also decompressing it to the original size [29]. Some works have tested this capability, adding the possibility to obtain values of the original data

for certain coordinates [9]. Other architectures have shown their capability learning implicit representations of a desired scene or geometry and being able, once trained, to recover the full original data by evaluating the neural network in the points of the original data [10]. Moreover, in these examples, the neural networks have shown the ability to hallucinate new unknown points, recovering more resolution than the original. In this work both examples can be understood as a solution to our problem but we will focus on the second approach and learn an implicit representation of the data using a multilayer perceptron architecture.

3.1.1 Multilayer perceptron

A perceptron [39] is a unique neuron defined as:

$$y = f\left(\sum_{i=1}^{n} x_i w_i + b\right), \tag{3.1}$$

where f is a non-linear activation function, like sigmoid or a hyperbolic tangent function, w are the weights of the perceptron, x are the inputs and b is the bias. This equation can be expressed in a matricial form:

$$y = f\left(\mathbf{x}\mathbf{w}^T + b\right). \tag{3.2}$$

Neurons can be organized forming layers. The number of neurons in each layer is decided depending on the number of the outputs of the layer since each neuron has only one output value. If multiple layers are connected like in the figure 3.1, the architecture is called multilayer perceptron. The number of layers and the number of neurons in each one are obtained experimentally as hyper-parameters except for the last layer, whose number of neurons is equal to the number of outputs values.

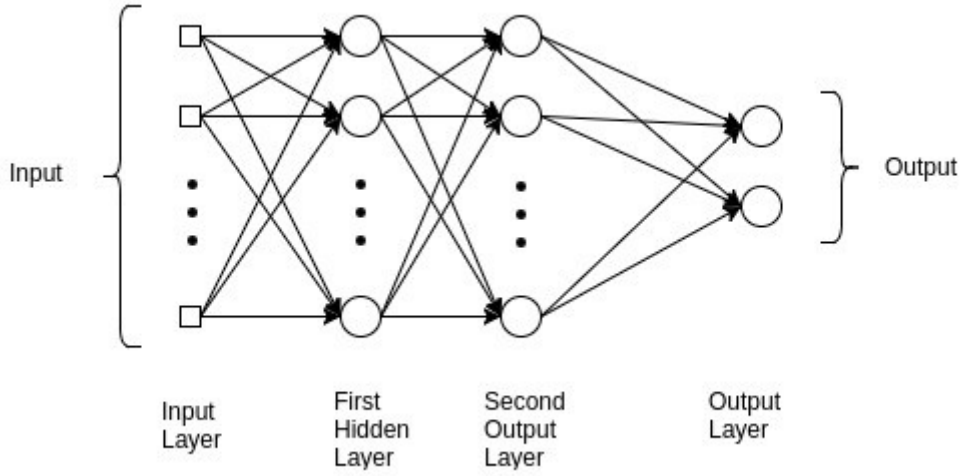

Figure 3.1: Multilayer perceptron scheme. Image from [40].

3.1.2 Activation functions

The capacity of the neural networks to solve difficult problems lies on the non-linearites that they are able to learn. They can learn them due to the non-linearites of its activation functions. The activation function basically transform the output of the neuron depending on certain conditions, which can change the whole distribution of the intermediate data. There are multiple activation functions such as hyperbolic tangent (equation 3.3), rectified linear unit (ReLU) [41] (equation 3.4), sigmoid (equation 3.5).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.3}$$

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{3.4}$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{3.5}$$

The type of function to use depends on the problem and the characteristics of the problem to solve. The selection of the activation function could be understood as an hyper-parameterization.. Despite the necessity of testing different activation functions, the number of candidates can be reduced based on the type of function that has been used for similar problems in previous works. For example, one of the most used activation functions to work with images is the ReLU function since it only propagates positives values. This is specially interestring when the output is an image because the standard range to work with them is in $[0,1]$. The characteristics of the activation functions help the networks to focus on the resolution of the problem. In contrast, selecting a bad activation function can diminish and hinder the convergence of the training process.

3.1.3 Gradient descent optimization

The optimization process is done through the minimization of a function. In deep learning the function to minimize is called the loss function and measures the performance of the network over a task. The perfect solution would be to find the global minimum of the loss. However, that could be impossible due to the complexity of the parameter space. On the other hand, finding a local minimum is an easier task that can be done with gradient descent optimization methods.

As a simplification, we can denote the loss function as $y = f(x)$ which derivative is $f'(x) = dy/dx$. It is known that the derivative gives the slope of f at the point x. With the direction of the slope we know in which direction the function will be minimized and if the derivative is equal to 0 then the function is in a minimum. f can be defined with several parameters in the form of $x = \{x_1, x_2, ..., x_n\}$. The derivative of f must be done with partial derivatives $\partial f(x)/\partial x_i$. The gradient of f denoted as $\nabla f(x)$ will be the vector with all the partial derivatives. Like the gradient gives the slope of every parameter we can update them in the negative direction of the slope and descending iteratively in the gradient. The updating function can be denoted as:

$$x_{t+1} = x_t - \varepsilon \nabla f(x_t), \tag{3.6}$$

where x_t is x values in the actual iteration and ε is a small value called learning rate that controls the speed of the descent. The learning rate can not be too high because when the function is near to the minimum a huge learn rate could prevent the function to find the local

minimum. On the other hand if the learning rate is too small the needed steps for getting the minimum point will be so high that the learning process would take too much time.

3.1.4 Stochastic gradient descent

The previous definition hold for a loss with a single training data. However, in machine learning is typical to have a hundred of thousands of training data. In that case, the loss function can be understood as a sum over the loss of all training data. Considering the parameters of the model as θ and any loss functions for a single data as L we can denote the loss $\mathcal{L}$ for all the example as

$$\mathcal{L}(\theta) = \mathbb{E}[L(x,y,\theta)] = \frac{\sum_{i=1}^{n} L\left(x^{(i)}, y^{(i)}, \theta\right)}{n}. \tag{3.7}$$

Then, to apply the gradient descent method is needed to calculate the gradient with respect to the parameters of the model θ as

$$\nabla_\theta \mathcal{L}(\theta) = \frac{\sum_{i=1}^{n} \nabla_\theta L\left(x^{(i)}, y^{(i)}, \theta\right)}{n}. \tag{3.8}$$

Computing the gradient descent with this calculation is one of the most successful options. However, as we commented above, the size of the dataset is huge and makes difficult training with the whole data due to memory space restrictions. The solution for that is training using a subset of data which is called batch size, and can be defined with any desired size. For the cases of really small batches they are called minibatches and for some fields they shown better results that bigger batches. Denoting the loss for these batches as $\mathcal{L}'$ and the number of elements in the batch as n' the gradient could be defined as be

$$\nabla_\theta \mathcal{L}'(\theta) = \frac{\sum_{i=1}^{n'} \nabla_\theta L\left(x^{(i)}, y^{(i)}, \theta\right)}{n'}. \tag{3.9}$$

Finally, the gradient descent for the parameters using batches is

$$\theta_t = \theta_{t-1} - \varepsilon \nabla_\theta \mathcal{L}'\left(\theta_{t-1}\right). \tag{3.10}$$

This variation of the gradient descent is called stochastic gradient descent and is one of the most used methods in the training of neural networks. As the original method, this one can not converge to the global minimum but, it will end in the nearest local minimum.

3.2 Non-line-of-sight (NLOS) transient imaging

Transient imaging methods leverage the information encoded in the temporal domain of a time-resolved light capture. One of the areas that can be addressed in this field is the non-line-of-sight (NLOS) reconstruction. The imaging methods developed in this area aim to reconstruct scenarios that are hidden around a corner, by analyzing their indirect illumination on a surface visible to the camera. A typical NLOS scene can be seen in figure 3.2 where an ultra-short laser pulse is emitted to a wall (also called *relay wall*), the light propagates from the hidden scene, being reflected by the objects, and part of it goes to the relay wall and is recorded by an ultra-high speed camera called single-photon avalanche diode (SPAD) generating a transient profile $\mathbf{H}$ called impulse response function. These profiles have the information of the hidden

scene encoded inside them. Thus, $\mathbf{H}$ is calculated for different points of the relay wall for the laser and the SPAD parameters are $\mathbf{H}(\mathbf{x}_l, \mathbf{x}_s, t)$ where $\mathbf{x}_l$ is the laser position, $\mathbf{x}_s$ is the sensor position and t a specific temporal instant.

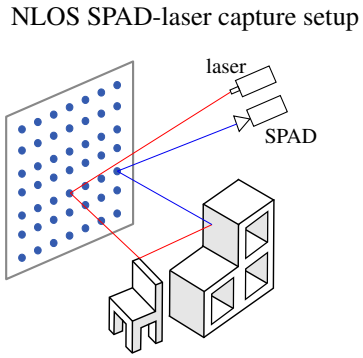

Figure 3.2: Sample setup for capturing a NLOS scene. An ultrashort light pulse is emitted to a relay wall and the sensor captures the light resolved in time reflected by the scene to the relay wall. Image adapted from Liu et al.[2].

The impulse response function $\mathbf{H}$ is used in several methods for reconstructing the hidden scene as we commented in section 2.1. In this work we focus in the phasor field framework [2] which is one of the best performing methods in NLOS imaging. This work gives us the possibility to transform the $\mathbf{H}$ function into a phasor field which transforms the temporal resolution into a complex magnitude at the relay wall with amplitude and phase dimensions for a concrete frequency.

3.2.1 Phasor fields

The work of Liu et al. [2] presents a new method for transient imaging which transforms the NLOS problem into a virtual line-of-sight (LOS) problem. This transformation allows the authors to use classical optic methods in the NLOS domain.

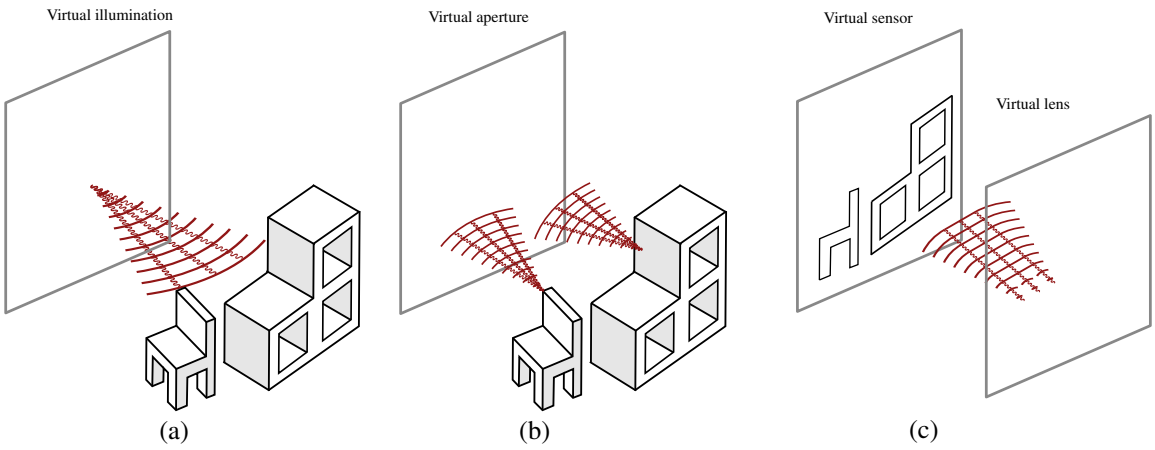

Figure 3.3: Phasor field steps. (a) Virtual illumination is propagated through the scene. (b) A virtual aperture capture the response of the scene to the virtual light. (c) A virtual lens focus the captured illumination of the virtual aperture and imaging. Image adapted from [2].

The linearity of $\mathbf{H}(\mathbf{x}_l, \mathbf{x}_s, t)$ is used by phasor fields to compute the response of the hidden scene (Fig. 3.3b) for any virtual complex-valued emission profile $\mathcal{P}(\mathbf{x}_l, t)$ (Fig. 3.3a) at points $\mathbf{x}_l$ in a virtual sensor as

$$\mathcal{P}(\mathbf{x}_s, t) = \int_L [\mathcal{P}(\mathbf{x}_l, t) * \mathbf{H}(\mathbf{x}_l, \mathbf{x}_s, t)] \, \mathrm{d}\mathbf{x}_l. \tag{3.11}$$

Through the propagation of the field $\mathcal{P}(\mathbf{x}_s, t)$ with an imaging operator $\mathrm{I}(\cdot)$ for any point $\mathbf{x}_v$ in the hidden scene as

$$\mathrm{I}(\mathbf{x}_v) = \Phi(\mathcal{P}(\mathbf{x}_s, t)). \tag{3.12}$$

This operator models a virtual lens and sensor system (Fig. 3.3c) and it can be formulated in terms of a Rayleigh-Sommerfeld diffraction propagator [2]. In the case of propagating monochromatic signals of a single frequency ω this image formation operator $\Phi(\mathcal{P}_\omega(\mathbf{x}_s, t))$ is denoted as

$$\Phi(\mathcal{P}_\omega(\mathbf{x}_s, t)) = \left| \int_S \mathcal{P}_\omega(\mathbf{x}_s, t) \frac{\mathcal{L}_\omega(\mathbf{x}_s, \mathbf{x}_v)}{|\mathbf{x}_v - \mathbf{x}_s|} \mathrm{d}\mathbf{x}_s \right|^2, \tag{3.13}$$

where $\mathcal{L}$ is a complex operator that changes the phase of $\mathcal{P}$ at frequency ω. For the Φ operator we implement it as a thin lens model that focuses into a virtual image plane at a hidden location $\mathbf{x}_v$, having

$$\mathcal{L}_\omega(\mathbf{x}_s, \mathbf{x}_v) = e^{-ik|\mathbf{x}_v - \mathbf{x}_s|}, \tag{3.14}$$

where $k = \omega/c$ is the wavenumber, with c the speed of light.

A position $\mathbf{x}_v$ can be imaging by combining the focused emission profile (Eq. 3.11) and the imaging operator (Eq. 3.13) as

$$I(\mathbf{x}_v) = \left| \int_S \int_L [\mathcal{P}_\omega(\mathbf{x}_l, t) * \mathbf{H}(\mathbf{x}_l, \mathbf{x}_s, t)] \frac{\mathcal{L}_\omega(\mathbf{x}_s, \mathbf{x}_v)}{|\mathbf{x}_v - \mathbf{x}_s|} \mathrm{d}\mathbf{x}_l \, \mathrm{d}\mathbf{x}_s \right|^2. \tag{3.15}$$

This method is fully computational since the unique external data that it need is the $\mathbf{H}$ function. Furthermore, the image formation model is virtual and can be formulated in other forms. The same idea holds for the emission profile and for this work we will define it as a constant-emission light source as

$$\mathcal{P}_\omega(\mathbf{x}_l, t) = e^{i\omega t}. \tag{3.16}$$

Chapter 4

Implicit representation of phasor fields

In the last decade, the field of non-line-of-sight (NLOS) has evolved improving the quality of the reconstructions of hidden scenes. However, the capture method has barely changed. NLOS methods use a captured impulse response function of a hidden scene. To obtain this function a light is emitted, by an ultra-short laser, to a diffuse wall (relay wall). Then it travels across the hidden scene and goes back to the relay wall where it is finally captured by a sensor. The capture process needs to be done sequentially for each point in the relay wall, resulting in a slow process with a bottleneck that forces to have impulses response functions with low spatial resolution.

Despite the problem with the spatial resolution, works as Liu et al. [2] successfully reconstruct high complexity hidden scenes. However, to obtain more detailed reconstructions, it is mandatory to generate more spatially dense impulse response functions. To obtain them there are two options: creating new hardware to increase the spatial resolution or using computational methods to generate new points from the captured ones.

On the one hand creating hardware to increase the spatial resolution of the sensor is not a trivial work. The actual sensors are single-photon avalanche diode (SPAD). They have a resolution of 1x1 pixel and there are some versions with a line of SPADs obtaining a row of captures. Despite these advances, the next generations of 2D SPADs will have a low resolution near 16x16 pixels which is insufficient to solve the problem. On the other hand, using fully computational methods can work with the actual capture systems. We can test and implement multiple algorithms or versions of them in a relatively short space of time. In spite of the advantages that computational methods can give us, the main limitation for us to design and create new capture hardware is the necessity of testing it with real capture systems which only a few laboratories in the world have. In contrast to test computational methods we can use both real and simulated data, and therefore, validate the method by using them. For that reason we have decided to implement a computational method and generate, through simulation, our capture data.

The problem that we want to alleviate is the low density of values in the spatial domain of the impulse response function to improve the quality of the reconstructions. We can say that this problem can be understood as an upsampling or superresolution problem where the number of samples of a function is increased by an algorithm. This problem is typical in images where the low resolution image is transformed into a high resolution one. The type of methods that gives better results in this domain are the machine learning based ones. However, to work with

this type of methods we have to take into account the characteristics of the data and its possible transformations.

The impulse response function contains the amount of photons (light) that the sensor receives in a time interval (in the order of nanoseconds). Light decays quadratically with distance causing the quantity of light captured by the sensor to decay rapidly with each bounce in the hidden scene resulting in a high dynamic range and strong changes of values with zeros in most of the space. Designing or using a machine learning method to deal with these characteristics will be complex. Furthermore, the temporal dimension is several orders of magnitude larger than the spatial one. Such unbalance could lead to the machine learning method focus more into time than space, achieving unsuccessful results. For those reasons, in this work we have decided to use the work by Liu et al. [2] to transform the impulse response function into a phasor field with multiple frequencies, resulting in two spatial dimensions and one frequency dimension (for more details about the characteristics of the impulse response function and its transformation see the section 4.2).

Once we have decided what type of data we will work with, we can look for machine learning methods to alleviate our problem. As we commented before, increasing the resolution or the number of samples in a function is a known problem. Some of the most successful methods to do this are the neural networks based ones due to their ability to hallucinate new samples. We can differentiate two types of methods. The first one uses a network to increase the resolution into a fixed size [8]. This method is general and mainly used with images because, once it is trained, it can be used with any type of image. The second one uses a network to learn the value of the samples of the function in their positions [10, 11, 12]. This is called implicit representation and it is a specific solution for a single function. Despite the necessity of training a network for each function, this approach has the advantage that the implicit representation can be sampled in a continuous space. In other words, an implicit representation is a transformation of a discrete domain (the training samples) into a continuous one. This means that we can obtain any desired resolution from it, in contrast with the general methods that only upsample to a fixed resolution. Other important difference between the two methods is that the first one requires a dataset with a size of hundreds of thousands, but the second one only uses the samples of the function that is learning. Since the amount of data is an important restriction in NLOS (generating thousands of captures is unpractical) and the second approach gives us the possibility to upsample in a continuous space, we have decided to follow this type of approach.

Specifically, in this work we will follow the ideas of the works of Mildenhall et al. [10] and Sitzmann et al. [12]. They use a neural network model called multilayer perceptron (MLP) to learn an implicit representation as we want. Despite the fact that they apply this model to different domains and for different purposes, we can get ideas and parts of them to solve the problems that we have found during this work. In our case, instead of working with standard images, we have a phasor field formed by complex values and our goal is to increase its resolution. This could be similar to the work by Mildenhall et al. [10] where the authors learn from a set of images to generate new points of view, hallucinating the unknown points. We aim, instead of generating points in a 3D spatial domain, to generate points in spatial and frequency domain. Since the type of problem is different, the final solution of Mildenhall et al. [10] can not be applied to our problem. In the work by Sitzmann et al. [12], the authors present a new type of activation function for learning implicit representations. The authors

test this function with images and video but they do not try to generate new samples. Besides that, their work is an excellent example of the capability of implicit representations and can give us methods to improve our results. In the following section we explain the experiments done in this work formulating the problem in a mathematical form and how we use the ideas in previous works to improve our results.

4.1 Neural implicit representation of phasor fields

In this work we look for a neural network model to learn an implicit representation of a discrete phasor field. Once we have selected the type of function that we want to fit (a phasor field) and the type of application (one model per function), we can now describe a pipeline for our work (Figure 4.1), which describes all the stages involved, from capturing the discrete function **H**, transforming it into a discrete phasor field, and the conversion of them by fitting a neural network into a continuous phasor field that can be reconstructed with more resolution than the original.

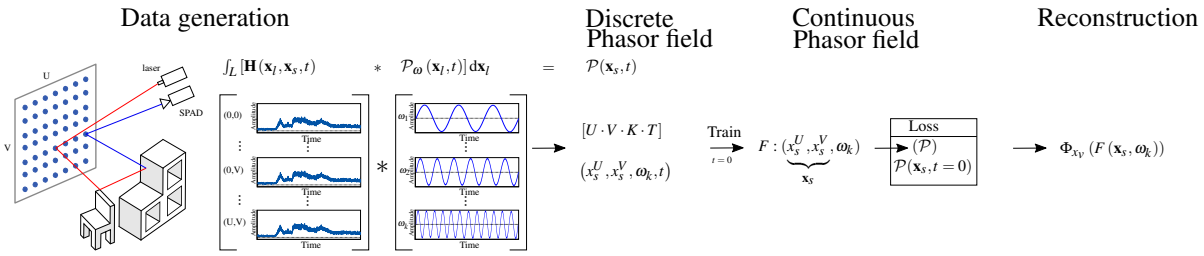

Figure 4.1: Pipeline used in this work. From left to right. Diagram of the capture setup. The discrete capture response impulse function **H** is convolved with different monochromatic waves with frequencies ω_i. k discrete phasor fields are obtained from the previous step giving a dataset with size $[U \cdot V \cdot K \cdot T]$ and input parameters $\left(x_s^U, x_s^V, \omega_k, t\right)$. For the training process we evaluate the dataset for $t = 0$. The model F is trained using as input the coordinates and frequency, and the loss is calculated with the original discrete function $\mathcal{P}$ obtaining a continuous version of the discrete $\mathcal{P}$. Once the model is trained, we can reconstruct at higher resolution or for new frequencies. Some parts of the figure are from Liu et al. [2]

After we defining the workflow, we can start to denote formally the problem. The discrete function can be denoted as

$$\mathcal{P}_\omega\left(\mathbf{x}_s\right) = F_{\omega,D} : (\mathbf{x}) \rightarrow (\mathbf{p}), \tag{4.1}$$

where D marks that the function is discrete, $\mathbf{x}$ is a vector with x, y discrete coordinates, $\mathbf{p}$ is a vector with two real values that correspond with the real and imaginary part of the phasor $\mathcal{P}$ for the position $\mathbf{x}$. And our implicit representation can be denoted as

$$\mathcal{P}_\omega\left(\mathbf{x}_s\right) \approx F_{\omega,M} : (\mathbf{x}) \rightarrow (\mathbf{p}), \tag{4.2}$$

where $\mathbf{x}$ is now in a continuous space. Our first approach is to use a multilayer perceptron (MLP) with hyperbolic tangent activations functions as our $F_{\omega,M}$, the M subindex denote that the function is an MLP model. For training this model and evaluating its performance as implicit representation we decided to train it with a phasor with resolution 32x32 and generate with it a phasor with an upsampled resolution of 64x64, in other words, the model is generating

a 75% of new points. The result for this case can be seen in figure 4.2. The MLP model learns the global structure but does not learn high frequency details.

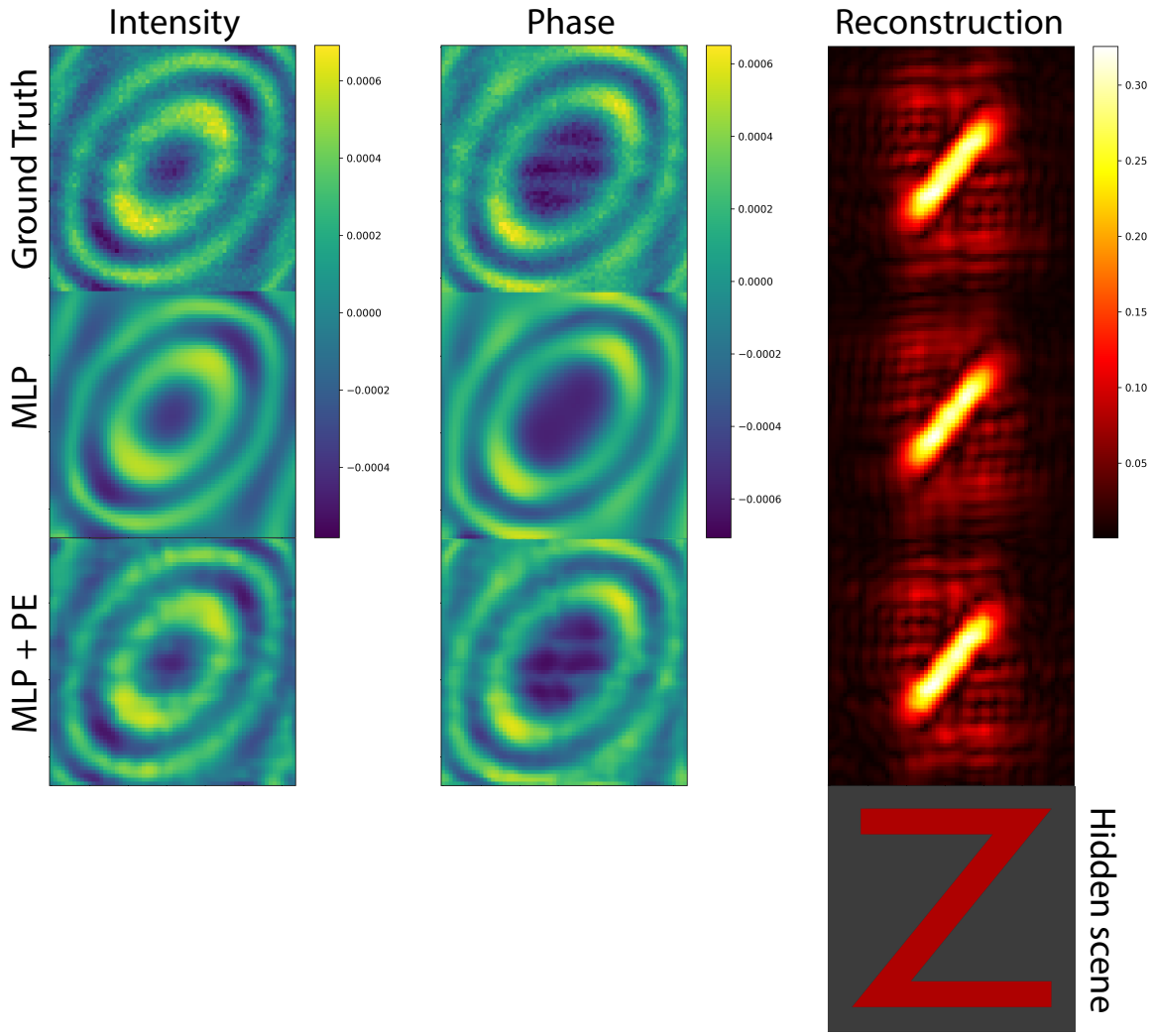

Figure 4.2: Results of an implicit representation model for the spatial dimensions from 32x32 to 64x64 resolution with a fixed frequency. Middle row: result of generating new points with an MLP model ($F_{\omega,M}$). Bottom row: result of generating new points with an MLP model transforming the input with positional encoding ($F_{\omega,P}$).

To solve this problem we have followed the work of Mildenhall et al. [10]. In this work, the authors presented a new method that allow a neural network to learn high frequency details by transforming the input values in a set of frequencies as

$$\gamma(\mathbf{x}) = \left(\sin\left(2^0\pi\mathbf{x}\right), \cos\left(2^0\pi\mathbf{x}\right), \cdots, \sin\left(2^{L-1}\pi\mathbf{x}\right), \cos\left(2^{L-1}\pi\mathbf{x}\right)\right). \tag{4.3}$$

This $\gamma(\mathbf{x})$ function is called by the authors as *positional encoding* and expands each input coordinate in a total of $2L$ frequencies. We can add this transformation to our formulation as

$$\mathcal{P}_\omega(\mathbf{x}_s) \approx F_{\omega,P} : (\gamma(\mathbf{x})) \rightarrow (\mathbf{p}), \tag{4.4}$$

where the subindex P denote the function with position encoding. In this work we set L considering 2^L to be as close as possible to the original resolution, in the case of a resolution

of 32 we have an $L = 5$. With this new approach we obtained the results that can be seen in figure 4.2. The MLP trained with this positional encoding gives more accuracy when learning the high frequency details than the first model, which learns a smooth version of the data.

The above results show how this method can be used to generate a continuous space of the input. Since the monochromatic phasor is defined for a concrete frequency, having an implicit representation that can generate the phasor for a continuous frequency space will increase the value of this method dramatically. Due to the implicit representation will learn the whole possible frequencies and automatically compute them without the necessity of doing convolutions for new frequencies (eq. 3.11) or saving the high dimensionality function $\mathbf{H}$. This new model can be denoted as

$$\mathcal{P}(\mathbf{x}_s) \approx F_P : (\gamma(\mathbf{x}), \omega) \rightarrow (\mathbf{p}), \tag{4.5}$$

where $\mathcal{P}$ is now a phasor for all possible frequencies and ω is now a parameter of the implicit representation. Hence now, we have a new dimension which make harder the challenge we are solving. Due to that, we decided to firstly limit it to only generate the implicit representation in the frequency domain. For that, we fix the resolution of the phasor to 64x64 and generate 151 frequencies and fit with 135 from the total. The results of this model can be seen in figure 4.3. Some frequencies are learned well, but for higher frequencies the results lose quality.

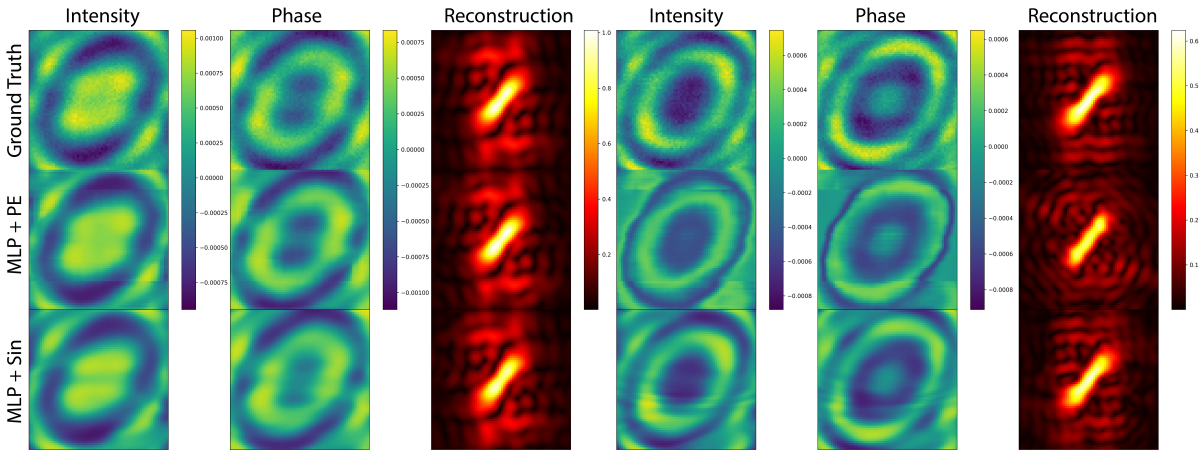

Figure 4.3: Results of two phasors with non-trained frequencies with fixed spatial resolution. Middle row: Phasors generated with a multilayer perceptron with positional encoding (F_P). Bottom row: Phasors generated with a multilayer perceptron with sin activations (F_S).

To solve this problem we follow the work of Sitzmann et al. [12]. The authors presented a new model for implicit representation that is able to learn high frequencies through changing the typical ReLU or Tanh activation functions with sine functions. This change allows the model to learn high frequency details without the necessity of a positional encoding. Using this new model and applying it to our problem we can denote it as

$$\mathcal{P}(\mathbf{x}_s, \omega) \approx F_S : (\mathbf{x}, \omega) \rightarrow (\mathbf{p}), \tag{4.6}$$

where the subindex S denotes the use of sine activation function in the model. With this new approach we obtained better results as can be seen in the bottom row of the figure 4.3. This

new model learns high frequency details and generate new middle frequencies correctly. Also, as the initial multilayer perceptron example (figure 4.2) this model removes the noise from the original data and has a few artifacts with form of wave in the middle-bottom part of the second phasor.

Once we have a model that can learn and generate phasors for new frequencies, we can use this model and try to generate a continuous space in all the dimensions by training it like in the previous case (fig. 4.2) with a spatial resolution of 32x32 and generating a space of 64x64. The results for this model can be seen in figure 4.4. This model shows a similar level of achievement than the previous one (figure 4.3) and has similar artifacts.

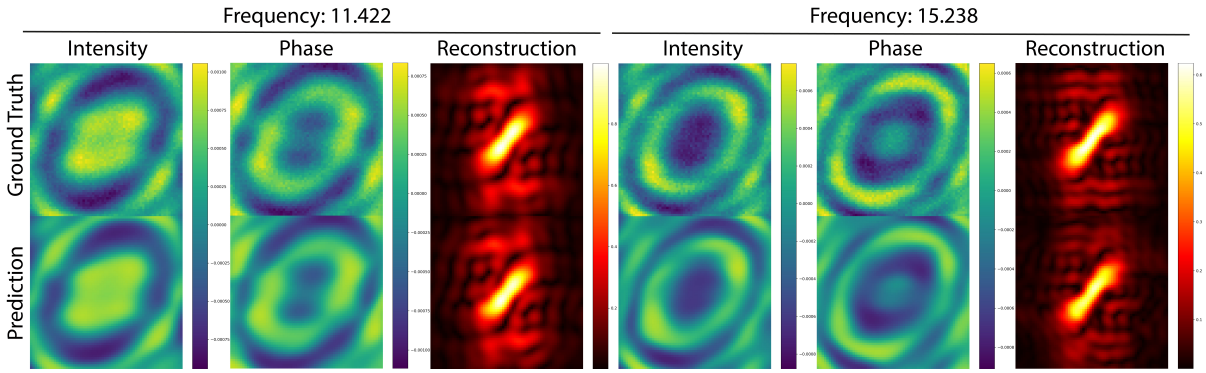

Figure 4.4: Results of two phasors with non-trained frequencies and with a spatial resolution of 32x32 increased by the multilayer perceptron with sine activations functions models (F_S) to 64x64.

Compression. A side effect of fitting a discrete function in a neural network is the capacity of learn, in fewer number of parameters, the whole amount of data of the discrete function at the same time that it generates a continuous input space. In the table 4.1 it can be seen how the different models learn the distribution of the discrete function in a fraction of its parameters. This case is specially strong for the multi-frequency models, where the percentage of parameters needed for fitting the original function is less than 1.5% of the total amount of the training values. We believe that the multi-frequency models are able to learn the phasor dataset with only four times more parameters, in contrast with the x135 values that conform the dataset with multi-frequency, because the model learn the internal structure of the phasor and how it changes with the different frequencies.

Experiment	**Parameters**	**Data size**	**Parameters/Data (%)**
Mono frequency + upsampling (MLP)	942	2048	31.57
Mono frequency + upsampling (MLP + PE)	914	2048	45.63
Multi frequency (MLP + PE)	3938	1105920	0.36
Multi frequency (MLP + Sin)	3362	1105920	0.31
Multi frequency + upsamppling (MLP + Sin)	3362	276480	1.22

Table 4.1: Compression rates of the different models tested. The compression column show what percentage of the training values are needed to represent them respect to the total amount of values of the dataset.

4.2 Data transformation

In machine learning, an important step is the study of the data and possible transformations on it. The main reason for analyzing it is to provide a more simple or processed data to the models and improve the convergence of them. Through the study of the data is possible to understand characteristics of it like how it is distributed, possible correlations, etc. This allows to select the better transformation and in some cases it makes possible to reduce the dimensionality of the data.

In our case, the data that we have is a captured response impulse function $\mathbf{H}(\mathbf{x}_l, \mathbf{x}_s, t)$. In the general case this function has two spatial and one temporal dimensions. The spatial dimensions $\mathbf{x}_l$ and $\mathbf{x}_s$ correspond to the coordinates of the laser and the SPAD sensor in the relay wall. Also, these spatial coordinates are in a 3D space but since the wall is in a parallel position with respect to the hidden scene the third dimension will remain the same and can be ignored. In total the $\mathbf{H}$ function has five dimensions.

As we commented previously chapter 1, our goal is to generate an implicit representation through neural networks. This implicit representation has as input the coordinates for the point that we want to sample, the same number of input dimensions as the original discrete function $\mathbf{H}$. This implies that the input of the model will be the five values of the coordinates and the output the value in that spatial and temporal point. To train a model with this we will need five times the total number of $\mathbf{H}$'s values only for the input of the model. In the table 4.2 can be seen how the size of the input change for a fixed standard temporal resolution. The sizes that the input reaches is too high even for lower resolutions. To solve this problem, we choose to work with a version of the $\mathbf{H}$ function that only captures a single point of camera in the center. By fixing this point $\mathbf{H}$ is now a 3D matrix and the input of the model have a workable size.

Attending to the values of the $\mathbf{H}$ function. In the figure 4.5 can be seen an example of the temporal profile for a single pair of laser-sensor. These profiles have a high dynamic range and the changes on its values are very abrupt. Due to these characteristics, the learning process is more difficult and models have low convergences.

Spatial Resolution	**Size in GB**
16	2
32	32
64	512
128	8192

Table 4.2: Size in memory of the input for training a five dimensional neural network model. The temporal dimension is fixed to a resolution of 4096.

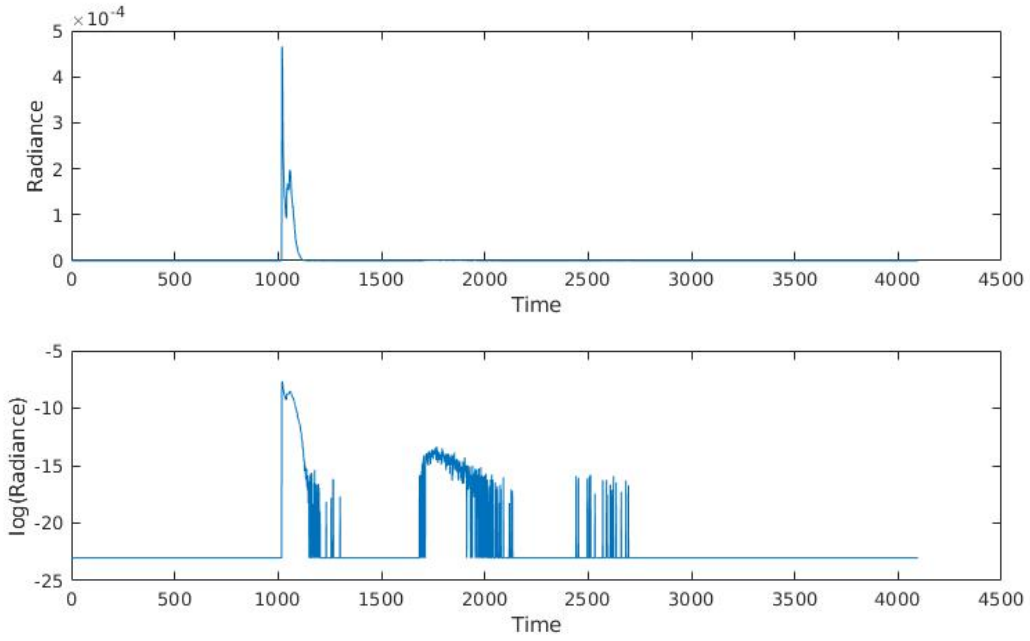

Figure 4.5: Example of **H** function for a concrete pair laser-sensor. The top image is the transient profile in its original scale. The bottom image is the transient profile in logarithmic scale.

To solve the problem with the high dynamic range of the **H** function, we propose to use the phasor fields framework (section 3.2.1) for NLOS reconstruction. In the first step of this framework the **H** function is convolved (eq. 3.11) with a virtual illumination phasor for a concrete frequency ω (eq. 3.16). This phasor $\mathcal{P}(\mathbf{x}_s,t)$ is a 3D complex matrix, two spatial dimensions and one temporal, but as the virtual light used in the convolution is constant along the time and the camera model is a conventional camera, $\mathcal{P}(\mathbf{x}_s,t)$ can be evaluated in $\mathcal{P}(\mathbf{x}_s,t=0)$ obtaining a 2D complex matrix and solving the problem with the high dynamic range. However, the frequency that modules the virtual illumination affects to the reconstruction quality as can be seen in figure 4.6. Furthermore, the correct frequency for each **H** function is not the same and needs to be obtained experimentally.

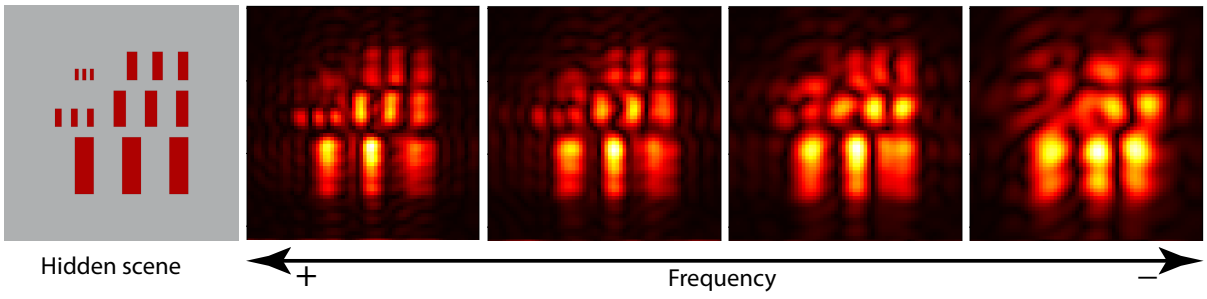

Figure 4.6: Example of different reconstructions with different frequencies.

Data normalization. In this work we have two types of data: the input parameters of the model (coordinates and in some models the frequency) and the values of the phasor which the model will learn. For both options we tested the $[0,1]$ and $[-1,1]$ normalization but the models

did not fit correctly in the borders of the matrix. Since the problem was in the boundaries we consider that the problem was with the normalization values in the limits of the range and with a normalization in $(-1,1)$ this problem was palliated.

Data generation. The capture of the **H** functions can be done from two different sources, real capture systems or through simulation. The first one can not be done in the laboratory at this university due to we do not have the hardware. However, using real captured data have some disadvantages. The captures have noise and their conditions can not be controlled completely. On the other hand, simulating avoids this problems due to the environment can be controlled (e.g., avoid external light contamination) and can be generated with less noise. Also, simulating allow to create idyllic conditions e.g., perfect lambertian surfaces for the relay wall. For that reason in this work we simulate all the captures using the public transient render by Jarabo et al. [13].

Generic model vs specific models. Neural networks are usually applied for generic applications that can be used once is trained with different data, for example in superresolution models, classification or some compression works. However, this models need to be trained with enormous datasets, in some cases millions of samples. In the context of this work this is not an option due to the amount of time that is needed to generate a dataset of the size that general models need. For that reason in this work we choose to use one model per function (or scene) that we want to fit.

Since we are fitting a single function in the neural model, and we want to infer new values, we need to avoid overfitting in the model. In other cases, if we wanted to completely compress a function and only evaluate at the original points, overfitting would be justified. Other aspect that is involved in the overfitting and in the capacity of the neural models to be able to fit the discrete function is the number of model parameters. The correct number of them needs to be obtained experimentally. Moreover, it can depend on the type of function that the model is fitting, complex functions can require more parameters than the simplest ones and if the number of parameters is too big, the model will be lazy and will overfit easily. In the context of this problem, we can consider an upper bound for the number of parameters, the total amount of values in the discrete function because it is the number of values needed to represent the same information. In our experiments we tested with different number of parameters and figured out that with more than half the total number of parameters of the original function, the models tend to overfit and can not generate well new points. On the other hand, with less than the half the models learn the discrete function and are able to generate new points smoothly. However, if the parameters are too low, the models can not converge at all.

Chapter 5

Results

In the section 4.1 we have tested different neural networks models to learn an implicit representation of a phasor field. Specifically, the neural network model obtained can learn an implicit representation of a multi-frequency phasor field. One of the advantages of learning a multi-frequency phasor field is the elimination of the necessity of calculating separate implicit representations for each frequency. This means that it gives the ability to compute any desired frequency that it is not in the sampled space. This is especially interesting due to the frequency is directly dependent of the quality of the reconstructions. Computing novel phasor fields with higher frequencies and higher grid densities can provide sharper results of the reconstructed geometry. As such, the ability of our implicit representation to compute novel temporal frequencies and spatial locations in the relay wall that were not provided by the sampled captured data is useful to provide better reconstructions. Another application of this multi-frequency model is the possibility to use it with more complex lights and cameras in the phasor field framework. To use them it will be required (theoretically) the use of infinite frequencies. With an implicit representation, this problem can be more treatable because the representation avoids the necessity to do any convolution for each frequency.

During the analysis of the different neural models (Section 4) we have tested them with only one scene for simplicity. After demonstrating that the MLP with sine activation functions model provides the best behavior in representing multi-frequency phasor fields, here we provide a deeper analysis of its performance in scenes of different complexity. However, before showing the different results, is important to understand the limitations of the virtual camera and the virtual light used in the phasor field framework. The virtual camera has an aperture size that corresponds to the size of the sampled grid on the relay wall. After using this aperture to simulate a virtual camera focused at certain depth in the hidden scene, all geometry behind or ahead of the focused plane is out of focus, introducing several artifacts in the image. Additionally, due to capture limitations, the virtual illumination is attenuated radially from the center of the relay wall, which results in attenuated reconstructions of the geometry further from the center of the resulting images. This effect is clearly visible in all the previous results in the section 4.1. To analyze how the implicit representations can approximate the discrete phasor field we will compare the intensity, the phase and the reconstruction of both phasor fields, discrete (ground truth) and the one predicted by the implicit representation. We have selected simple scenes to have low out-of-focus geometry to avoid out-of-focus artifacts. Reconstructing the in-focus geometry in more complex scenarios would require estimating a large amount of different phasor fields for each voxel of the reconstructed scene, while simple scenes provide good results with a single multi-frequency phasor field. We regard as future work to provide

more complex implicit representations for phasor fields in cluttered scenes.

As in the last model of the section 4.1, we will test the capacity of the model to generate new samples in the spatial and frequency domain. To train and test the model we have decided to remove 75% of the samples from a phasor field with a spatial resolution of 64x64, therefore getting only 32x32 sampled points in the relay wall. For the frequency, we have computed 151 different frequencies and trained with the 90% of them, reserving the last 10% for testing. Once the model is trained with the low resolution phasor field, we recover the original resolution at specific frequencies and spatial locations in the relay wall that the model has not been trained with.

For testing the model, a multilayer perceptron with sine activation functions, we have used four different scenes (figures 5.1-5.4). We start by a planar but detailed scene (figure 5.1). This scene allows us to analyze the performance under different geometric resolutions allowing seeing the level of details that the method is able to recover. Concretely, the results obtained for this scene show the capability of the model to learn more complex phasor fields and upsample them correctly.The general structure is maintained and an important denoised effect is appreciable. The reconstructions are qualitatively similar. They lose a bit of detail but, bearing in mind that the model is trained with the 25% of the original points, it is a good reconstruction. As comparison, in the figure 5.2 can be seen how is a reconstruction of a phasor field with a resolution of 32x32 and a reconstruction with a resolution of 64x64. In the error map it can be seen two things: the error derived from the noise not learned by the model and how the network fails more in the low values.

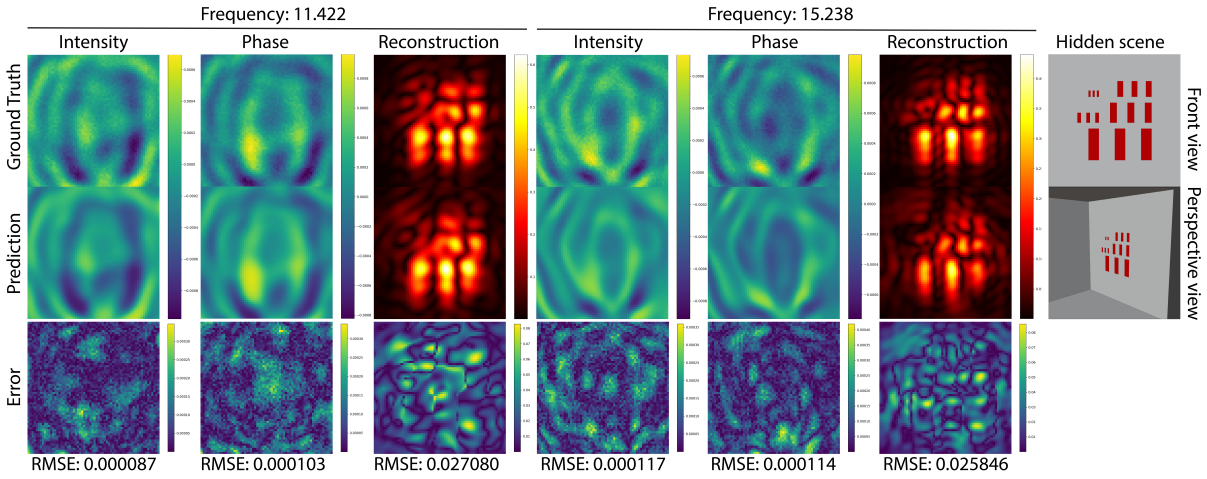

Figure 5.1: Results of recovering a scene with multiple planes with different sizes. The results show that the implicit representation learn correctly the shape of the phasor fields and remove the noise of them. The reconstructions are qualitatively similar. Left phasor field has a frequency of 11.422 and the rigth one has a frequency of 15,238.

The second scene is formed by a plane with several concavities and irregularities (figure 5.3). Note how the resulting amplitude and phase of the phasor field change due to the structural differences of the hidden geometry. Nevertheless, our implicit representation is able to estimate the phasor field structure at higher resolutions, very close to the ground truth while also removing noise, and provides a similar reconstruction result. Our implicit representation

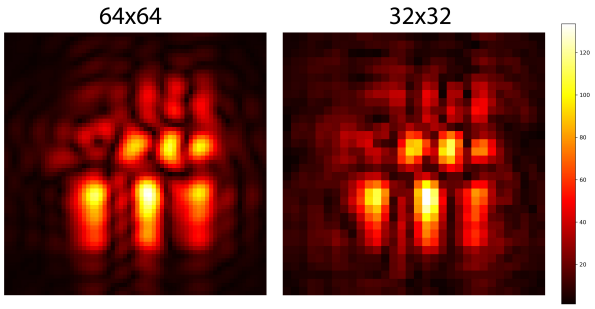

Figure 5.2: Comparison of the reconstruction of the same scene with different resolution, 32x32 and 64x64 for the same frequency (15.238).

is capable of learning the general structure, while simultaneously removing the noise in the amplitude and phase images. The error maps show, as in the previous example, that the model has more error in the low value points and also the general noise error. The reconstructions are very similar. Despite that the RMSE in the first reconstruction is higher, the result is qualitatively more similar than in the second reconstruction.

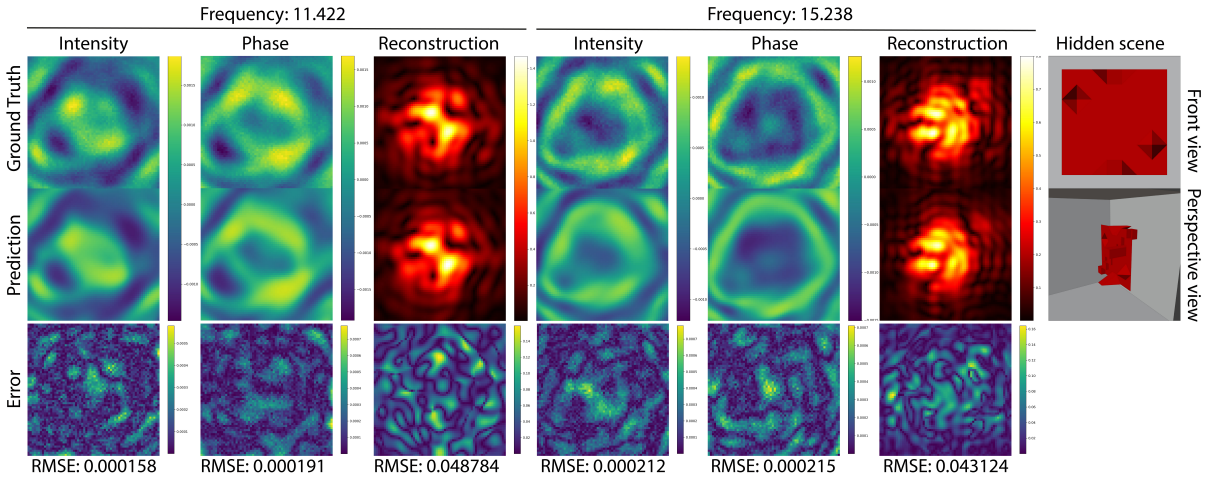

Figure 5.3: Results for a scene with multiple concavities and pikes. The reconstructions do not show the shape correctly due to the out-of-focus geometry. The neural model learns and recover the shape of the phasor field removing the noise from it.

Finally, we have two similar scenes (figures 5.4 and 5.5). Both are formed by two rectangular planes at two different depths but in each scene the distance between them is different. The goal of these two scenes is to test the performance of our algorithm under the presence of objects that may occlude themselves from the perspective of the virtual camera. For these two cases our implicit neural model learns the structure of the phasor fields as well.

Comments and discussions. Our neural representation of phasor fields has shown a high capacity to learn the general structure of the phasor field, being able to generate higher resolutions (recovering a 75% of the points in these tests) and new frequencies with similar results to the ground truth. The RSME is similar in all of them but is bigger in the phasor field with higher frequency. This could be caused by the increase of details that appear when the temporal modulation frequency is increased. This increase in the temporal frequency results in

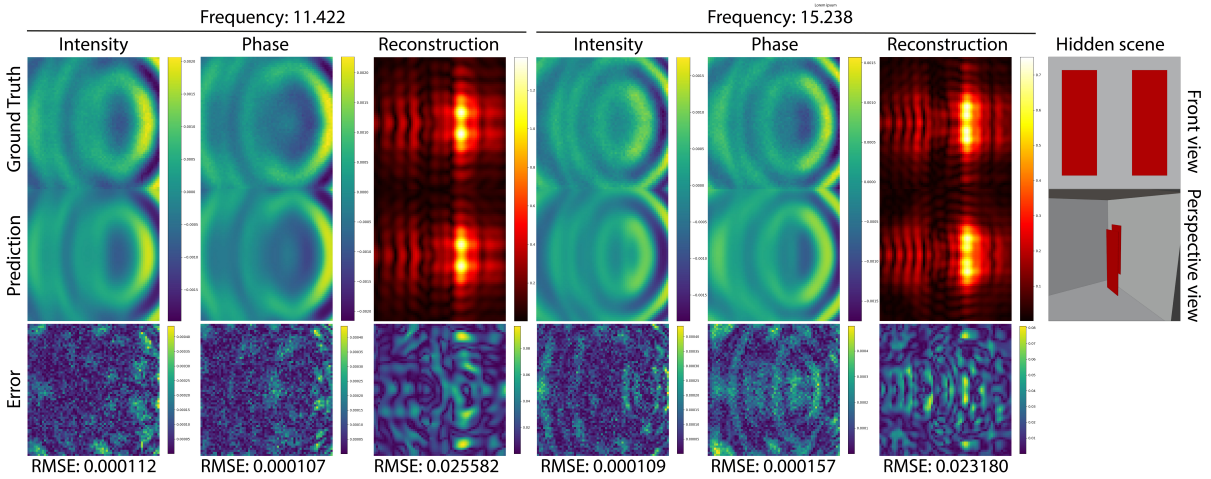

Figure 5.4: Results of a scene with two planes at different depth. This scene allows us to test if the phasor field with multiple objects and depths can be learned by the neural model. The model learns correctly despite the changes in the shape.

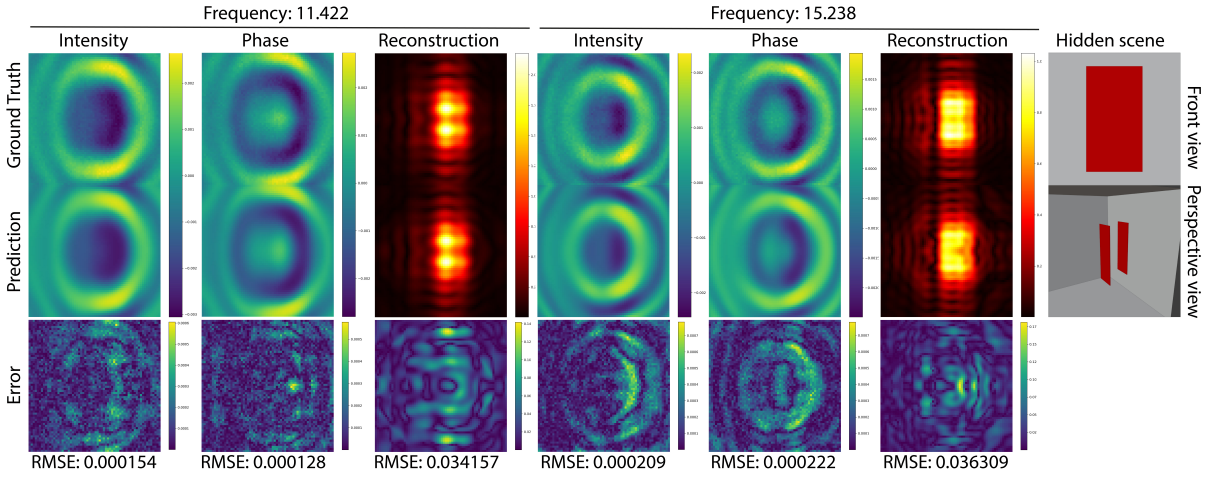

Figure 5.5: Results of a scene with two planes at different depth. In this case the plane in the back is semioccluded by the plane in the front. The neural model learns correctly the shape and even the strong change of values.

an increment of the spatial frequency in the phasor field, as the results show. Focusing on the reconstructions, the predicted and the ground truth are also very similar and can recover even the artifacts produced by diffraction effects. Note that this is a problem inherent in the phasor fields method [2], and not a result of our neural model. The biggest qualitative differences between predicted and ground truth appear in the reconstructions done with the phasor fields with higher frequencies, losing more details than with the low frequency phasor fields. Note that our quantitative comparisons are done with respect to the ground truth solution with equivalent sampling density and temporal frequency. While our model yields a larger RMSE when increasing modulation frequency, increasing sampling density and modulation frequency in phasor fields provides better reconstructions of the hidden scene, as our qualitative results show. The differences in RMSE of our model with respect to lower frequencies arise from the ability of our model to analytically represent higher frequencies, and not from the ability of the data to properly reconstruct the scene.

Hidden scene	**Loss (L2)**	**Training time (s)**
Figure 5.1	0.009883426	677.92
Figure 5.3	0.011194097	670.95
Figure 5.5	0.0029691448	673.33
Figure 5.4	0.0032978356	673.98

Table 5.1: Losses and training times for each scene. The training process have been done with a learning rate of 0.001 and a total of 100000 epochs.

The model in all our tests has shown a denoising ability. This effect is produced by the capability of neural networks to focus on predictable aspects of the data. Concretely, as the noise is something more random than the structure of the data, the neural networks obtain better results learning the structure and characteristics of the data. However, this effect is only observed when the parameters of the networks are not too high because, in that case, the neural network overfits the data and cannot upsample. Regarding the possible effect of the noise in the error maps, we thought that it is possible that the noise could mask the correct level of achievement of this method. However, to test it, it would be necessary to use other metrics that take more into account the structure of the nearby pixels as the structural similarity index measure (SSIM). We will leave the test of this hypothesis for a future work. In the reconstructions, the effect of the existence of noise in the phasor field does not appear to be determinant or at least the effect is too low to be seen. In their error maps, the first difference regarding the error maps of the phasor field is the lack of noise. The error maps of the reconstructions are more smoothly than the error maps of the phasor fields. We thought that the reconstruction process is resistant to this level of noise. With this idea, a future work that could be done is an analysis of the tolerance of the reconstruction method at different levels of noise and how this method can be used for denoising it. This is an interesting aspect because the real captures have noise that usually needs to be added to the synthetic data to compare both. Furthermore, knowing that, it could help to know the level of noise that this method can handle and use it to generate synthetic data faster.

The training process of these scenes is fast (see table 5.1 for exact training times). In around eleven minutes each scene can be learned. The evaluation process for each frequency is practically instant, less than one second, giving a huge potential to use it for more applications such as using more complex cameras that could require computing thousands of different frequencies. Moreover, the model can be evaluated with multiple frequencies at the same time without a perceptible increase of time.

Chapter 6

Conclusions and future work

In this work we have introduced a new method to represent phasor fields in an implicit form by using neural models to increment the efficiency and the quality of the results of reconstructions of hidden scenes. Phasor fields are a recent work that allows to take virtual photographs of a hidden scene as it was seen from a relay wall. However, this method is limited by in its resolution caused by the capture process. Since each spatial point has to be sequentially captured, it becomes unpractical and complex to capture high resolution information. Although this problem could be alleviated with the development of new hardware, this approach would require the use of expensive hardware configurations.

Inspired by other works on implicit representations that prove their ability to create a continuous space from discrete number of samples. We have formulated a phasor field using implicit representation which allows us to learn a discrete phasor field and sample it in a continuous space. This transformation removes any resolution limitation. In other words, we could obtain infinite samples from it. Following recent studies and works, we have tested different models for different tasks. Starting by the most simple (learning an implicit representation for a single-frequency phasor field without changing the resolution), up to the most complex (that learns an implicit representation of a multi-frequency phasor field). This last model is a multilayer perceptron with sine activation functions that can upsample the data in the spatial and frequency domains.

To verify if the final model can generalize to different scenes we have tested it with four scenes. The model correctly upsample the data for all the scenes giving a similar error in all of them. Other aspect observed is the denoising effect, since the model recovers smooth phasor fields with the correct structure. This can be qualitatively demonstrated, as the reconstructed scenes are almost identical to the ground-truth ones.

This work left some interesting research as future avenues. Our current method could be extended to support more cluttered scenes with more complex imaging functions, where each location of the scene would require a different illumination and lens function, leading to the estimation of higher number of phasor fields per scene. An aspect that can be tested is whether the noise in the captures is related with the differences in the reconstructions. Related to the noise, it could be interesting to analyze how the noise affects to the reconstructions and if this model or other types of network can denoise the phasor field and with which level of noise are they effective. Moreover, this work could be expanded to the study of the whole four dimensions involved in phasor fields, in contrast to the two explored in this work. Another

idea that can be useful in the future work is training the model end-to-end. This would require developing a differentiable implementation of the phasor field framework. By training the model directly over the reconstruction, instead of directly over the phasor field, better results could be obtained, allowing the model to learn better the spatial relationships of the samples. Finally, this work is the first to our knowledge, to leverage machine learning techniques under the phasor fields framework. A further development of this idea could lead to methods that could help the field of NLOS to obtain better results.

Bibliography

[1] Adrian Jarabo, Belen Masia, Julio Marco, and Diego Gutierrez. Recent advances in transient imaging: A computer graphics and vision perspective. *Visual Informatics*, 1(1), 2017.

[2] Xiaochun Liu, Ibón Guillén, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Adrian Jarabo, Diego Gutierrez, and Andreas Velten. Non-line-of-sight imaging using phasor-field virtual wave optics. *Nature*, 572(7771):620–623, 2019.

[3] Kazunori Akiyama, Antxon Alberdi, Walter Alef, Keiichi Asada, Rebecca Azulay, Anne-Kathrin Baczko, David Ball, Mislav Baloković, John Barrett, Dan Bintley, et al. First m87 event horizon telescope results. iv. imaging the central supermassive black hole. *The Astrophysical Journal Letters*, 875(1):L4, 2019.

[4] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[5] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *arXiv preprint arXiv:2001.05566*, 2020.

[6] Chunwei Tian, Yong Xu, Lunke Fei, and Ke Yan. Deep learning for image denoising: a survey. In *International Conference on Genetic and Evolutionary Computing*, pages 563–572. Springer, 2018.

[7] Amlaan Bhoi. Monocular depth estimation: A survey. *arXiv preprint arXiv:1901.09402*, 2019.

[8] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. pages 4799–4807, 2017.

[9] Gilles Rainer, Abhijeet Ghosh, Wenzel Jakob, and Tim Weyrich. Unified neural encoding of btfs. *Computer Graphics Forum (Proceedings of Eurographics)*, 39(2), June 2020.

[10] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*, 2020.

[11] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *arXiv*, 2020.

[12] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

[13] Adrian Jarabo, Julio Marco, Adolfo Muñoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. A framework for transient rendering. *ACM Trans. Graph.*, 33(6), 2014.

[14] Ahmed Kirmani, Tyler Hutchison, James Davis, and Ramesh Raskar. Looking around the corner using transient imaging. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[15] Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G. Bawendi, and Ramesh Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature Communications*, (3), 2012.

[16] Victor Arellano, Diego Gutierrez, and Adrian Jarabo. Fast back-projection for non-line of sight reconstruction. *Optics Express*, 25(10), 2017.

[17] Martin Laurenzis and Andreas Velten. Nonline-of-sight laser gated viewing of scattered photons. *Opt. Eng.*, 53(2), 2014.

[18] Wenzheng Chen, Fangyin Wei, Kiriakos N. Kutulakos, Szymon Rusinkiewicz, and Felix Heide. Learned feature embeddings for non-line-of-sight imaging and recognition. *ACM Trans. Graph.*, 39(6), 2020.

[19] Shumian Xin, Sotiris Nousias, Kiriakos N Kutulakos, Aswin C Sankaranarayanan, Srinivasa G Narasimhan, and Ioannis Gkioulekas. A theory of Fermat paths for non-line-of-sight shape reconstruction. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 6800–6809, 2019.

[20] Chia-Yin Tsai, Aswin C. Sankaranarayanan, and Ioannis Gkioulekas. Beyond volumetric albedo – a surface optimization framework for non-line-of-sight imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[21] Xiaochun Liu, Ibón Guillén, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Adrian Jarabo, Diego Gutierrez, and Andreas Velten. Non-line-of-sight imaging using phasor fields virtual wave optics. *Nature*, 2019.

[22] David B Lindell, Gordon Wetzstein, and Matthew O'Toole. Wave-based non-line-of-sight imaging using fast fk migration. *ACM Trans. Graph.*, 38(4):1–13, 2019.

[23] Marco Renna, Ji Hyun Nam, Mauro Buttafava, Federica Villa, Andreas Velten, and Alberto Tosi. Fast-gated 16× 1 spad array for non-line-of-sight imaging applications. *Instruments*, 4(2):14, 2020.

[24] Subrahmanyan Chandrasekhar. *Radiative Transfer*. Dover, 1960.

[25] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers Inc., 1994.

[26] Adrian Jarabo. Femto-photography: Visualizing light in motion. Master's thesis, Universidad de Zaragoza, 2012.

[27] Stephan Meister, Rahul Nair, Bernd Jähne, and Daniel Kondermann. Photon mapping based simulation of multi-path reflection artifacts in time-of-flight sensors. Technical report, Heidelberg Collaboratory for Image Processing, 2013.

[28] Julio Marco, Quercus Hernandez, Adolfo Muñoz, Yue Dong, Adrian Jarabo, Min Kim, Xin Tong, and Diego Gutierrez. Deeptof: Off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Trans. Graph.*, 36(6), 2017.

[29] Yun Liang, Mingqin Chen, Zesheng Huang, Diego Gutierrez, Adolfo Muñoz, and Julio Marco. Compression and denoising of transient light transport. *Optics Letters*, 2020.

[30] Miguel Galindo, Julio Marco, Matthew O'Toole, Gordon Wetzstein, Diego Gutierrez, and Adrian Jarabo. A dataset for benchmarking time-resolved non-line-of-sight imaging. In *IEEE International Conference on Computational Photography (ICCP), posters*, 2019.

[31] Maik Keller, Jens Orthmann, Andreas Kolb, and Valerij Peters. A simulation framework for time-of-flight sensors. In *International Symposium on Signals, Circuits and Systems 2007*, 2007.

[32] Maik Keller and Andreas Kolb. Real-time simulation of time-of-flight sensors. *Simulation Modelling Practice and Theory*, 17(5), 2009.

[33] Matthias B Hullin. Computational imaging of light in flight. In *SPIE/COS Photonics Asia*, 2014.

[34] Javier Grau Chopite, Matthias B. Hullin, Michael Wand, and Julian Iseringhausen. Deep non-line-of-sight reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[35] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122, 2016.

[36] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018.

[37] Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. Neural btf compression and interpolation. *Computer Graphics Forum (Proceedings of Eurographics)*, 38(2), March 2019.

[38] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019.

[39] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[40] *Multilayer perceptron scheme*, 2020 (accessed December 2, 2020). `https://medium.com/@AI_with_Kain/understanding-of-multilayer-perceptron-mlp-8f179c4a135f`.

[41] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.