Designing Freeform Optics with Physics Informed Machine Learning

Literature Review

Marek Oerlemans, MSc

Delft University of Technology, The Netherlands

February 11, 2022

TUDelft

TUDelft

Freeform lens design

What is a freeform lens?

Figure: Example of a freeform optical element use for projecting the president of the USA

Design decisions

- Optical design
- Network structure
- Parametric design
- Sources

Optical theory

- Start with Maxwell's equations
- Assume nice properties and an ansatz for the solution → Helmholtz equation

$$\left(\nabla^2 + k^2\right) U = 0 \tag{1}$$

- Use Huygens principle → Rayleigh-Sommerfeld integral

$$U(x, y, z) = \frac{1}{j\lambda} \int_{\text{aperture}} U(x', y', 0) \frac{z}{r} \frac{\exp(jkr)}{r} dx' dy', \tag{2}$$

Solving the RS integral

Rayleigh-Sommerfeld integral

$$U(x, y, z) = \frac{1}{j\lambda} \int_{\text{aperture}} U(x', y', 0) \frac{z}{r} \frac{\exp(jkr)}{r} dx' dy',$$

Three options

- Fresnel: approximate the radius $\left(r \approx z + \frac{x^2+y^2}{2z} + \frac{x'^2+y'^2}{2z} - \frac{xx'+yy'}{z}\right) \rightarrow$ Fourier transform valid in the near field
- Fraunhofer: approximate the radius $\left(r \approx z + \frac{x^2+y^2}{2z} - \frac{xx'+yy'}{z}\right) \rightarrow$ Simpler Fourier transform valid in far field
- Rayleigh Sommerfeld: $U(x, y, z) = \mathcal{F}^{-1}\{\mathcal{F}\{U(x, y, 0)\} G(\alpha, \beta, z)\}$
    - More computationally expensive
    - More valid

Adding a lens

Adding a lens

Adding a lens has the simple effect that the phase becomes longer dependent on the thickness of the lens. That is, if the phase is ψ before the lens, the lens is Δ thick, then the phase directly behind the lens is $\psi + \Delta$

Convolutional neural networks

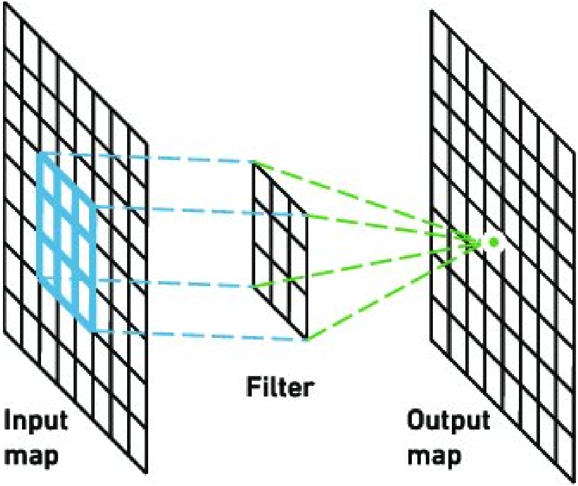

Figure: Convolutional layer

Neural network

- Fully connected layer
- Convolutional layer
- Max-pooling layer
- Batch normalization
- Residual connection

Residual connection

Also known as skip connection

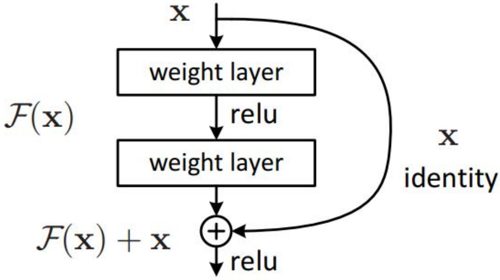

Figure: Example of a residual layer

Properties

- Convolutional neural networks
- Helps with vanishing gradients

NURBS

- B-splines: polynomial basis functions
- NURBS: divide a b-spline by a b-spline → rational function
- B-splines vs NURBS:
  - NURBS have double the parameters (in our case)
  - NURBS can approximate more curves: famous example is a circle

$$S(u, v) = \sum_{j=1}^{m} \sum_{i=1}^{n} N_{i,p}(u) M_{j,p}(v) P_{i,j} \tag{3}$$

vs

$$S(u, v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) M_{j,p}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p} M_{j,p}(v) w_{i,j}} \tag{4}$$

Literature

- Freeform Design
- Diffractive Neural Networks
- Phase Retrieval

Freeform design

- Monge-ampere equations
- Minimization problem
- Applications
  - LED street lights
  - Lens with specific field-of-view
- Continuous lens does not allow for discontinuous patterns

Diffractive Neural Networks

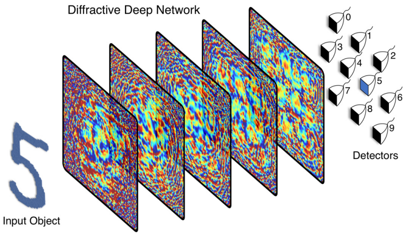

Figure: Diffractive Neural Network

- Diffractive Optical Elements
- Optimized digitally with RS diffraction
- A show of possibilities

Phase Retrieval

Phase retrieval

Finding the correct phase change for a given target intensity pattern

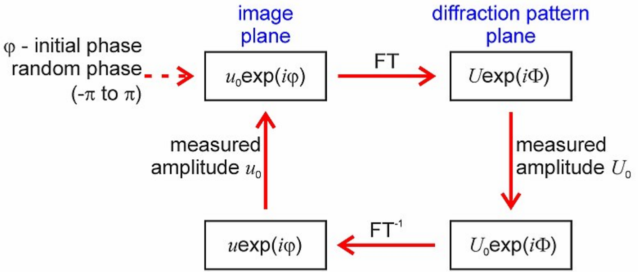

Figure: The Gerchberg-Saxton iterative algorithm for phase retrieval

Properties:

- Always descending error
- Similar to gradient descent on a specific loss function

Deep Phase Retrieval

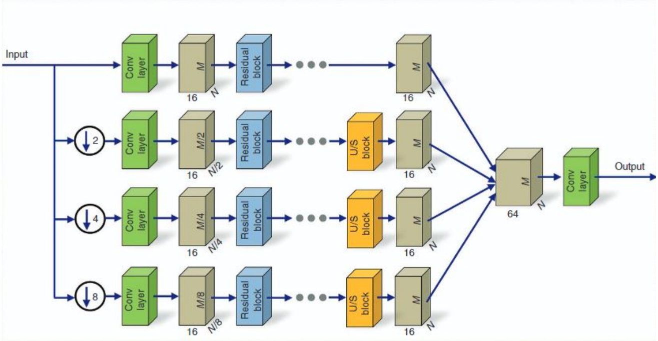

Figure: The network structure used for phase and amplitude recovery [1]

[1] Rivenson et al. (2018)

Previous work

- Joost
  - One Dimensional
  - Multiple Images
  - Convolutional Neural Network
  - Three Control Points
- Lucas
  - Two Dimensional
  - One Image
  - Small Fully Connected Network
  - 20 by 20 Control Points

Proposed Research

- Continue the work of Joost and mainly Lucas

Ultimate goal

Make neural network that takes as input the target intensity profile and the source intensity profile and outputs the NURBS parameters for the freeform lens that would make this possible

Intermediate steps

- Phase Retrieval Network
- NURBS network
- Possible second lens
- See if we can train one network with source as input vs train a network for each source

Phase Retrieval Network

Convolutional neural network similar to U-net

- Input: Target intensity (and possibly source intensity)

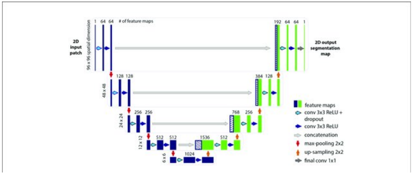

Figure: U-net network architecture

- Output: Same dimension phases
- Loss: Squared/absolute difference between formed NURBS surface and target surface (self-supervised)

TUDelft

Deep Phase Retrieval

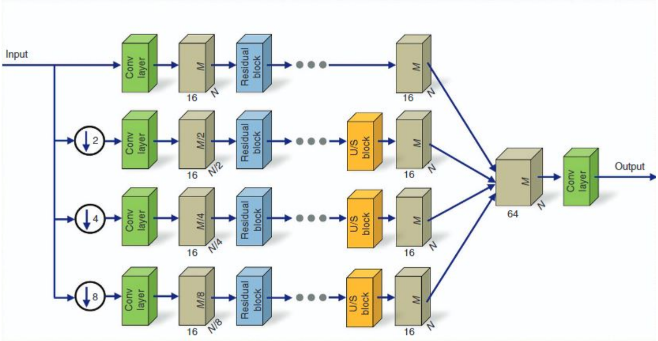

Figure: The network structure used for phase and amplitude recovery [1]

[1] Rivenson et al. (2018)

U-net

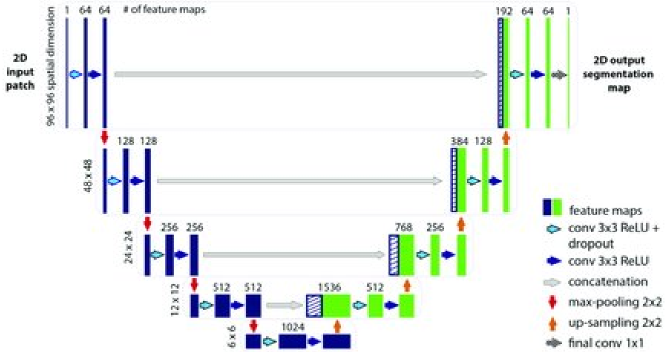

Figure: U-net network architecture

NURBS network

Convolutional neural network

- Input: Surface to be estimated
- Output: NURBS parameters that form the surface
- Loss: Squared/absolute difference between formed NURBS surface and target surface (self-supervised)

Proposed Research

- Continue the work of Joost and mainly Lucas

Ultimate goal

Make neural network that takes as input the target intensity profile and the source intensity profile and outputs the NURBS parameters for the freeform lens that would make this possible

Intermediate steps

- Phase Retrieval Network
- NURBS network
- Possible second lens
- See if we can train one network with source as input vs train a network for each source