

Towards Fast and Accurate Computational Algorithms for Vision Correcting Displays

Joshua Chen

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-110

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-110.html>

May 16, 2024



Copyright © 2024, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Towards Fast and Accurate Computational Algorithms for Vision Correcting Displays

by Joshua Chen

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Brian A. Barsky

Professor Brian Barsky
Research Advisor

15 May 2024

(Date)

Avideh Zakhor

Professor Avideh Zakhor
Second Reader

5/16/24

(Date)

Towards Fast and Accurate Computational Algorithms for Vision Correcting Displays

Copyright 2024
by
Joshua Chen

Abstract

Towards Fast and Accurate Computational Algorithms for Vision Correcting Displays

by

Joshua Chen

Master of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Brian Barsky, Chair

Millions of people across the world have visual aberrations that prevent them from using their digital devices without corrective eyewear. Vision correcting displays aim to present an in-focus image to the user without the use of such eyewear. This work proposes two new methods for performing computational vision correction. The first method builds upon existing research utilizing compressive sampling for image deconvolution. The second method utilizes a Vision Transformer-based model to perform image deconvolution. These two methods are presented and evaluated against previous methods. Lastly, future research directions are suggested that could improve upon the methods in this work and bring vision-correcting displays closer to a practical application that millions of people can use.

To my friends and family and their continued support.

Contents

Contents	ii
List of Figures	iv
List of Tables	v
1 Introduction	1
1.1 Optometry Background	1
1.2 Problem Description	2
2 Related Work	3
2.1 Algorithms for Conventional Displays	3
2.2 Light Field Displays	3
2.3 Algorithms for Light Field Displays	4
3 Compressive Sampling	8
3.1 Conventional Sampling	8
3.2 Reconstructing from Compressed Measurements	9
3.3 Sampling with Structurally Random Matrices	10
4 Compressive Deconvolution	11
4.1 Motivation	11
4.2 Problem Setup	12
4.3 Compressive Deconvolution with Frequency Sub-Banding	12
4.4 Evaluation	13
5 Deconvolution with Vision Transformers	23
5.1 Motivation	23
5.2 Methodology	23
5.3 Experiments	27
6 Future Work	32
6.1 Faster Compressive Deconvolution	32

6.2	Improving Vision Transformers	32
6.3	Higher-Order Aberrations	33
7	Conclusion	34
	Bibliography	35

List of Figures

1.1	Anatomy of the human eye [12]	2
4.1	Compressive deconvolution with frequency sub-banding	13
4.2	Target images for compressive deconvolution with frequency sub-banding experiments	14
4.3	Retinal projections of compressive deconvolution prefilters with and without frequency sub-banding	19
4.2	Comparison of PSNR for compressive deconvolution with and without frequency sub-banding	20
4.3	Comparison of SSIM for compressive deconvolution with and without frequency sub-banding	21
4.4	Comparison of perceptual loss (VGG-16 layer relu2.2) for compressive deconvolution with and without frequency sub-banding	22
5.1	Overview of Vision Transformer-based prefiltering. Although the input and pre-filtered image are shown in color, for a multi-channel color image, each channel is prefiltered separately, similar to compressive deconvolution.	25
5.2	Transformer encoder layer using <i>spatially separable self-attention</i> (SSSA)	27
5.3	Visual comparison of Point-to-Point, Many-to-Many, Area-to-Area, and Transformer methods. The Transformer method shown was trained with MAE + relu1.1 VGG-16 perceptual loss (weighted).	30

Acknowledgments

I would like to thank Professor Brian Barsky for continuing to run the vision correcting display project, and for encouraging me to explore my interests within the project. I would also like to thank Anmol Parande for introducing me to compressive deconvolution and helping me develop a better understanding of the vision correcting display problem. I also want to thank Saketh Malyala and Matthew Fogel for working with me on the vision correcting display project. Finally, I would like to thank Professor Avidesh Zakhor for teaching me signal and image processing in EE 120 and EECS 225B and for reviewing this work as a second reader.

Chapter 1

Introduction

In 2021, an estimated 63.7% of adults in the United States wore prescription eyeglasses [9]. The number of people with visual aberrations is likely greater than that. Without visual correction, vision for these people is blurry and out-of-focus. A *vision correcting display* (VCD) would correct for a user's visual aberrations, allowing them to view and interact with the device without needing external corrective eye-wear. These devices could help improve convenience and safety for users. For example, with a VCD, a far-sighted user would be able to clearly look at their GPS while driving without needing to get their glasses. Additionally, users would be able to use AR/VR headsets without needing to wear glasses, thus improving their comfort. This work extends the work by Parande in [20], improving on the accuracy and runtime of algorithms for such displays.

1.1 Optometry Background

In order for a user to see an image, the eye must capture and focus light from the outside world (see fig. 1.1). First, light hits the cornea, which is the outermost clear layer of the eye. The cornea acts as a lens with a fixed focal length and contributes approximately two-thirds of the eye's refractive power [22]. As light is refracted by the cornea, it travels through the pupil to the lens. The focal length of the lens is adjusted to refract the light onto the retina [17], where rod and cone photoreceptor cells convert the light into electrical signals which the brain uses to see an image.

The *near point distance* is the closest distance which the eye can focus at, and conversely, the *far point distance* is the furthest distance which the eye can focus at. The near point distance for a typical human eye is considered to be at 25 cm, but for those with far-sightedness (hyperopia/presbyopia), the near point distance is further than 25 cm. Similarly, the far point distance for a typical human eye is considered to be at infinity, but for those with near-sightedness (myopia), the far point distance is finite. These visual aberrations occur when the focal lengths of the cornea and lens do not match with the length of the eye, causing light to focus too far in front of or behind the retina. Astigmatism is caused

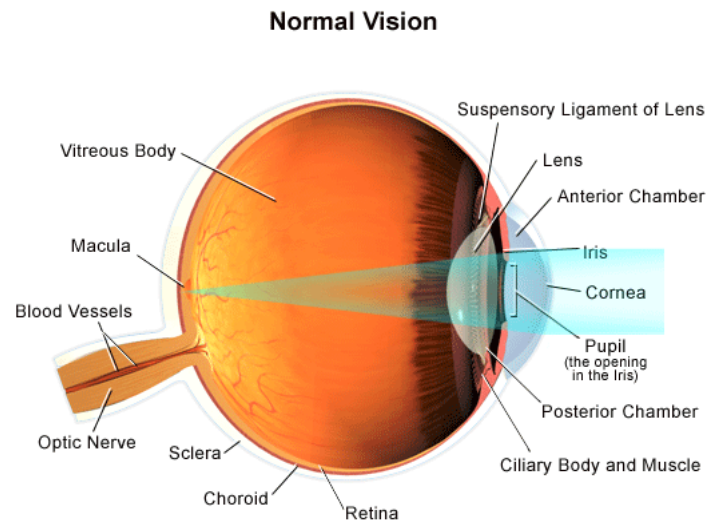


Figure 1.1: Anatomy of the human eye [12]

by an asymmetric shape of the cornea or lens and results in blurry vision at all distances. These “lower-order” aberrations can be corrected with eyeglasses. However, 15% of visual problems are caused by “higher-order” aberrations which cannot be corrected by eyeglasses [24]. For both lower- and higher-order aberrations, a VCD would computationally correct for a user’s visual aberrations such that the image appears in-focus without the need for external eye-wear.

1.2 Problem Description

This work aims to address the following problem:

Can we design a display such that, given measurements of a user’s visual aberrations, the displayed image would appear in-focus to the user without the need for external corrective eye-wear?

Mathematically, if y is the in-focus image that we would like the user to see, and $f(\cdot)$ represents the propagation and refraction of light from the display to the user’s retina, we would like to determine x such that $f(x) \approx y$.

This work specifically focuses on far-sightedness (hyperopia/presbyopia), since far-sighted individuals have a difficult time viewing their digital devices, which are typically placed close to the user, making this a common use case. However, in theory, the algorithms presented in this work should be applicable to other visual aberrations as well.

Chapter 2

Related Work

2.1 Algorithms for Conventional Displays

On a conventional display, the intensity of a pixel is purely a two-dimensional function of position $i(x, y)$ ¹. We can model the blur of a user’s eye as a convolution with a point spread function (PSF) $k(x, y)$. Thus, the image that the user would see $y(x, y)$ can be written as a convolution

$$y(x, y) = i(x, y) * k(x, y) \quad (2.1)$$

If we have some in-focus image $y^*(x, y)$ that we would like to user to see, then the problem is equivalent to deconvolving $y^*(x, y)$ and $k(x, y)$. However, since a blur is a low-pass filter, solving the deconvolution problem is ill-posed. In [13], Huang demonstrates deconvolution results using both frequency-domain and spatial-domain solvers, such as the Wiener filter and Richardson-Lucy solver. However, regardless of the method, the perceived image always has ringing artifacts, low contrast, or is still blurry. In order to address these issues, Huang proposed using a *light field display* instead of a conventional display.

2.2 Light Field Displays

A *multilayer display* is a display consisting of stacks of semi-transparent light-emitting panels separated by small gaps. In [14], Huang et al. showed that with multilayer displays, each layer has its own PSF with different non-overlapping zero-crossings in their optical transfer functions, thus preserving the frequency content of the prefiltered image and allowing the user’s eye to act as an all-pass filter rather than a low-pass filter. By using a multilayer display, the contrast of the perceived image is enhanced, and ringing artifacts are eliminated.

Because multilayer displays require increasing the thickness of the display, a light field display was proposed as a means of displaying *virtual* layers at different distances. The intensity of a pixel on a light field display is a four-dimensional function of both its position

¹We treat each channel as a separate grayscale image.

and the angle at which it is viewed $i(x, y, u, v)$. In practice, light field displays can be constructed by placing a pinhole array or a microlens array over a high resolution display. In this work, “display pixel” will refer to a pixel on the high resolution display, and “macropixel” will refer to a single pinhole or microlens which covers multiple display pixels.

2.3 Algorithms for Light Field Displays

Optimization-Based Algorithms

In optimization-based algorithms, a matrix P is constructed which models the relationship between the display and the image on the user’s retina. Once this matrix is constructed, a least-squares problem is solved in order to prefilter the target image. Since P is large and often ill-conditioned, in practice, the least-squares problem is solved iteratively using L-BFGS-B [1] rather than the closed-form solution.

Light Field Projection

In [15], Huang et al. constructed a matrix P where for a given discretized light field on the display \mathbf{f} , the image on the user’s retina \mathbf{i} would be given by

$$\mathbf{i} = P\mathbf{f} \quad (2.2)$$

Thus, for an in-focus image \mathbf{i} , the optimal light field to display would be given by the solution to

$$\min_{\mathbf{f}} \|\mathbf{i} - P\mathbf{f}\|_2^2 \text{ s.t. } 0 \leq \mathbf{f} \leq 1 \quad (2.3)$$

A detailed algorithm for the construction of P can be found in [20].

Forward Method

Because light field displays constructed with pinhole or microlens arrays have a conventional display underneath, Wu introduced the *forward method* in [26], which directly solves for the optimal display pixels rather than the display light field. Similar to [15], a matrix P is constructed where for a given image on the display \mathbf{x} , the image on the user’s retina \mathbf{i} would be given by

$$\mathbf{i} = P\mathbf{x} \quad (2.4)$$

In order to construct P , light rays are sampled from the display to the user’s retina, and P_{ij} indicates the number of light rays that start at display pixel j and hit retinal pixel i . Finally, each row of P is normalized to sum to 1 in order to keep image brightness constant. A detailed algorithm for the construction of P can be found in [20].

In order for this method to be accurate, the *one-to-one assumption*, which states that each display pixel is visible through at most one pinhole, must hold. Zhen shows in [29] that

a sufficient condition for the one-to-one assumption is

$$\left(\frac{d(n-2)}{f_l} + 1\right)p \geq a \quad (2.5)$$

where

- d := distance from the display to the aperture plane
- n := width of one macropixel (measured in display pixels)
- p := width of one display pixel
- f_l := distance from the display to the pinhole mask
- a := aperture diameter

For the experiments in this work, where each macropixel covers 5×5 display pixels, each display pixel has width 0.078 mm, the pinhole mask is 6 mm away from the display, and the aperture has diameter 3 mm, the one-to-one assumption is valid when the user is at least 75 mm away from the display.

Ray-Tracing Algorithms

Ray-tracing algorithms sample rays from the display to the user’s retina. However, unlike optimization-based algorithms, there is no matrix or least-squares problem that is constructed. Instead, a heuristic method is applied in order to directly set the color of each display pixel. Therefore, these algorithms tend to produce lower-quality results than optimization-based algorithms. However, because ray-tracing is highly parallelizable and no iterative optimization is required, these methods are faster than optimization-based algorithms.

Many-to-Many and Point-to-Point Algorithms

The Many-to-Many algorithm, similar to the forward method, samples light rays from the display to the user’s retina [29]. It is called *many-to-many* because for each display pixel, many light rays are sampled, resulting in many starting and many ending points per display pixel. However, unlike the forward method, rather than constructing a matrix and solving an optimization problem, instead, the value of each display pixel p is set to the average values of the pixels in the target image that were hit by light rays sampled from p . This heuristic approach means that the value of each display pixel can be computed in parallel since they do not affect each other. However, this results in a lower-quality image when viewed by the user.

The Point-to-Point algorithm, proposed by Yue in [27], is similar to the Many-to-Many algorithm. However, unlike the Many-to-Many algorithm, the Point-to-Point algorithm only

samples a single light ray originating from the center of each display pixel. This is equivalent to the Many-to-Many method with a sampling rate of 1. This reduces the amount of computation required to prefilter each image, at the cost of image quality.

The distinction between these two algorithms and the forward method can be summarized by the following two points:

1. When a sampled light ray does not land exactly on an integer-valued index of the retinal pixels, the Many-to-Many and Point-to-Point algorithms perform bilinear interpolation of the target image pixel values. On the other hand, the existing implementation of the forward method performs nearest neighbor interpolation.
2. Let P be the matrix constructed in the forward method, and let Q be equal to P before its rows have been normalized. By construction, each column Q_i of Q sums to 1 (or 0 if the light from display pixel i does not hit the retina). Then for a target $n \times n$ image \mathbf{i} , the computed prefilter \mathbf{x}^* of the Many-to-Many and Point-to-Point algorithms can be written as

$$\mathbf{x}^* = Q^T \mathbf{i} \quad (2.6)$$

In general, Q^T is not a right inverse of P and only serves as an approximation for the pseudo-inverse. However, when the number of sample light rays per display pixel is 1, then this approximation is exact.

Proof. Assume the number of sample light rays per display pixel is 1. In other words, each column of Q has at most 1 nonzero entry. Since each column of Q sums to either 0 or 1 by construction, then every entry in Q must be either 0 or 1. Therefore,

$$Q_{ij} = \begin{cases} 0 & \text{if } P_{ij} = 0 \\ 1 & \text{if } P_{ij} > 0 \end{cases} \quad (2.7)$$

Since by construction, each row of P sums to 1, then $(PQ^T)_{ii} = 1$. Additionally, since every column of Q has at most 1 nonzero entry, then for all k ,

$$Q_{jk} = 1 \implies \forall i \neq j, Q_{ik} = 0, P_{ik} = 0 \quad (2.8)$$

Therefore, $(PQ^T)_{ij} = 0$.

Thus, $(PQ^T)_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$, and $PQ^T = I$. □

Area-to-Area Algorithm

In the Area-to-Area algorithm proposed by Zhen in [29], rather than sampling many rays across a single display pixel, instead, each display pixel is mapped to a continuous area on the retina. This is done by tracing a ray from each corner of the display pixel to the user's

retina. This gives the boundary for a continuous region in the target image. Then, the value of each display pixel p is set to the average values of the pixels in the target image that were within the boundary of the continuous region that was mapped onto from p . This aims to effectively achieve a higher sampling rate than the Point-to-Point algorithm without the computational complexity of the Many-to-Many algorithm.

Chapter 3

Compressive Sampling

3.1 Conventional Sampling

In signal processing, sampling, sometimes also called sensing, is the process of reducing a continuous-time signal to a discrete-time signal. Conventionally, *uniform sampling* is used, where one measurement is taken every T units of time. T is known as the *sampling interval* or *sampling period*, and $f_s = 1/T$ is known as the *sampling rate* or *sampling frequency*. Mathematically, if we have a one-dimensional continuous-time signal $x(t)$, then our sampled signal is given by $\tilde{x}[n] = x(nT)$. If $x(t)$ contains no frequencies greater than B , then by the Nyquist-Shannon sampling theorem, $x(t)$ is uniquely determined by $\tilde{x}[n]$ if $f_s > 2B$ [21].

However, uniform sampling is just one possible sampling scheme. In general, given some functions $\{\phi_1(t), \dots, \phi_m(t)\}$, we can sample $x(t)$ by measuring

$$\tilde{x}_i = \langle x, \phi_i \rangle, \quad i = 1, \dots, m \quad (3.1)$$

Under this framework, uniform sampling is performed when $\phi_i(t) = \delta(t - iT)$, where $\delta(t)$ is the Dirac delta. In other words,

$$\delta(t) = 0 \text{ for } t \neq 0 \quad (3.2)$$

$$\int_b^c \delta(t) dt = 1 \text{ for } b < 0 < c \quad (3.3)$$

Although this framework can be applied to continuous-time signals, this work focuses on sampling discrete-time signals $\mathbf{x} \in \mathbb{R}^n$. This is because on digital devices, both the in-focus images that we would like the user to see and the images that we can display on the screen are discrete-time signals. Additionally, although images are two-dimensional, in this work, they will be represented as a single column-vector, where the columns of the image have been concatenated together.

Thus, the framework above can be rewritten as

$$\tilde{\mathbf{x}} = \Phi \mathbf{x} \quad (3.4)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the original signal we would like to sample, $\tilde{\mathbf{x}} \in \mathbb{R}^m$ is the sampled signal, and Φ is an $m \times n$ matrix representing m elements of the measurement basis. If $m \ll n$, then $\tilde{\mathbf{x}}$ is a compressed measurement of \mathbf{x} .

3.2 Reconstructing from Compressed Measurements

Let Ψ be an $n \times n$ matrix, where the columns ψ_1, \dots, ψ_n form an orthonormal basis. \mathbf{x} can be written in the basis Ψ as $\mathbf{x} = \Psi\mathbf{a}$. If \mathbf{a} is S -sparse (i.e., \mathbf{a} has at most S nonzero entries), then \mathbf{x} can be reconstructed by $\mathbf{x}^* = \Psi\mathbf{a}^*$, where \mathbf{a}^* is the solution to the convex optimization problem

$$\min_{\mathbf{a}} \|\mathbf{a}\|_1 \text{ s.t. } \tilde{\mathbf{x}} = \Phi\Psi\mathbf{a} \quad (3.5)$$

Candes and Romberg showed in [3] that the probability of exact reconstruction exceeds $1 - \delta$ if

$$m \geq C\mu^2(\Phi, \Psi)S \log\left(\frac{n}{\delta}\right) \quad (3.6)$$

where C is some positive constant, and $\mu(\Phi, \Psi)$ is the *coherence* between the measurement basis Φ and the representation basis Ψ . The coherence measures the largest correlation between any two elements of Φ and Ψ and is given by

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq i, j \leq n} |\langle \phi_i, \psi_j \rangle| \quad (3.7)$$

When $\mu(\Phi, \Psi)$ is small, Φ spreads out the information of \mathbf{x} in the Ψ basis and allows for the number of measurements m to be quadratically smaller. Furthermore, eq. (3.6) implies that only $\mathcal{O}(\log n)$ measurements are required to reconstruct a signal of length n .

However, in practice, most natural images are not S -sparse. Instead, their wavelet transforms often have many small nonzero coefficients which can be zeroed out with little perceptual loss to image quality. Nevertheless, Candès showed in [4] that if $\Phi\Psi$ satisfies the $2S$ -restricted isometry property (RIP) with *restricted isometry constant* $\sqrt{2} - 1$, then we can still retrieve the S largest coefficients by solving eq. (3.5). Mathematically, define the *isometry constant* δ_S for $\Phi\Psi$ as the smallest number such that

$$(1 - \delta_S) \|\mathbf{z}\|_2^2 \leq \|\Phi\Psi\mathbf{z}\|_2^2 \leq (1 + \delta_S) \|\mathbf{z}\|_2^2 \quad (3.8)$$

holds for all S -sparse vectors \mathbf{z} . δ_S measures how well $\Phi\Psi$ preserves lengths of S -sparse vectors, and δ_{2S} measures how well $\Phi\Psi$ preserves distances between pairs of S -sparse vectors. If $\delta_{2S} < \sqrt{2} - 1$, then

$$\|\mathbf{a}^* - \mathbf{a}\|_1 \leq C_0 \|\mathbf{a} - \mathbf{a}_S\|_1 \quad (3.9)$$

and

$$\|\mathbf{a}^* - \mathbf{a}\|_2 \leq \frac{C_0}{\sqrt{S}} \|\mathbf{a} - \mathbf{a}_S\|_1 \quad (3.10)$$

where \mathbf{a}_S represents \mathbf{a} with all but the largest S entries set to zero, and C_0 is some positive constant.

Furthermore, as stated by Candès and Wakin in [2], if the bases represented by Φ and Ψ are orthonormal, then with high probability, it is sufficient to have

$$m \geq C \cdot S \cdot \log^4 n \quad (3.11)$$

where C is some positive constant. Although there exist other matrices that satisfy the RIP with overwhelming probability provided that

$$m \geq C \cdot S \cdot \log \left(\frac{n}{S} \right) \quad (3.12)$$

such as the Gaussian matrix [2], because these matrices are dense and take up a lot of memory, they are impractical for large images.

3.3 Sampling with Structurally Random Matrices

Do et al. introduced the *Structurally Random Matrix* (SRM) for compressed sensing in [10]. The SRM can be constructed as the product of three matrices

$$\Phi = \sqrt{\frac{n}{m}} DFR \quad (3.13)$$

- R is an $n \times n$ randomizer matrix which is either a uniform permutation matrix or a diagonal random matrix whose entries are i.i.d. Bernoulli random variables with equal probability (i.e., $\mathbb{P}(R_{ii} = \pm 1) = 1/2$).
- F is an $n \times n$ orthonormal transform matrix, such as the FFT or DCT.
- D is an $m \times n$ subsampling matrix which selects a random subset of rows from FR .

Since each of these three matrices has a fast implementation that does not require any matrix multiplications, multiplication by the SRM is fast and does not require a large amount of memory. Furthermore, the SRM is a universal measurement matrix, meaning that sensing performance is equally good with almost all representation bases Ψ .

The probability of exact recovery with the SRM is at least $1 - \delta$ if

$$m \geq \mathcal{O} \left(S \log^2 \left(\frac{n}{\delta} \right) \right) \quad (3.14)$$

If $S > 16 \log(2n/\delta)$, then it is sufficient for

$$m \geq \mathcal{O} \left(S \log \left(\frac{n}{\delta} \right) \right) \quad (3.15)$$

Chapter 4

Compressive Deconvolution

Chapter 3 discussed compressive sampling of some signal $\mathbf{x} \in \mathbb{R}^n$. This chapter will discuss previous work applying compressive sampling to the deconvolution problem $\mathbf{y} \approx P\mathbf{x}$, as well as a proposed improvement upon previous work.

4.1 Motivation

Although the optimization-based algorithms discussed in section 2.3 provide accurate results, they are significantly slower than the heuristic ray-tracing algorithms for two main reasons, which makes them impractical since they cannot be run in real-time. One reason they are slow is because of the large size of the matrix P used by those algorithms. For an $n \times n$ target image and $u \times u$ display pixels per macropixel, P maps from a u^2n^2 display image to an n^2 retinal image, resulting in $n^2 \times u^2n^2$ -sized matrix. Not only does this impact the runtime of the L-BFGS-B solver, simply constructing this matrix itself takes a significant amount of time as well. Thus, reducing the size of P could potentially improve the runtime of any optimization-based algorithm.

Compressive sampling provides a method of using a wide measurement matrix Φ in order to sample and recover a signal \mathbf{x} . However, in the VCD problem, we do not have access to the optimal display image \mathbf{x} , so we cannot compute its measurements $\Phi\mathbf{x}$. Instead, we only have the target image \mathbf{y} and the relationship between the display and retina P . Nevertheless, we could instead compute the target image's measurements $\Phi\mathbf{y}$ and try to find the display image \mathbf{x} such that $\Phi\mathbf{y} \approx \Phi P\mathbf{x}$. Parande demonstrated in [20] that this method of *compressive deconvolution* could be applied to the VCD problem.

4.2 Problem Setup

The least-squares optimization problems in section 2.3 could be modified into compressive deconvolution problems by

$$\min_{\mathbf{x}} \frac{1}{2\mu} \|\Phi\mathbf{y} - \Phi P\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_p^p + \gamma \|\Psi^{-1}P\mathbf{x}\|_1 \quad \text{s.t. } 0 \leq \mathbf{x} \leq 1 \quad (4.1)$$

where μ , λ , and γ are hyperparameters, and $1 \leq p \leq 2$ controls the type of regularization on \mathbf{x} . An explanation of each term in the objective function is provided below.

- $\frac{1}{2\mu} \|\Phi\mathbf{y} - \Phi P\mathbf{x}\|_2^2$ is similar to the typical ordinary least-squares objective function $\frac{1}{2\mu} \|\mathbf{y} - P\mathbf{x}\|_2^2$. However, rather than penalizing errors in pixel-space, errors are instead computed in the measurement space given by Φ .
- $\lambda \|\mathbf{x}\|_p^p$ regularizes against noise in \mathbf{x} . $p = 1$ assumes that \mathbf{x} is corrupted by Laplacian noise, as in the Bayesian interpretation of LASSO regression. $p = 2$ assumes that \mathbf{x} is corrupted by Gaussian noise, as in the Bayesian interpretation of ridge regression. In the VCD problem specifically, the main source of noise is discretization error. However, in practice, it is considered to be negligible, and this term is ignored ($\lambda = 0$).
- $\gamma \|\Psi^{-1}P\mathbf{x}\|_1$ is the typical sparsity penalty used in LASSO regression. In this case, the image on the user's retina $P\mathbf{x}$ is encouraged to be sparse in the representation basis Ψ .

Eq. (4.1) is solved using the Alternating Method of Direct Multipliers (ADMM) algorithm, since it only requires ΦP , rather than Φ and P separately.¹ This is important since, as discussed in section 3.2, if the bases represented by Φ and Ψ are orthonormal, then with high probability, it is sufficient for Φ to have $\mathcal{O}(n^2 \log^4 n)$ entries. If ΦP could be computed directly, then only $\mathcal{O}(u^2 n^2 \log^4 n)$ entries would need to be computed, rather than the $\mathcal{O}(u^2 n^4)$ entries required for P . However, as in [20], the SRM is used as Φ , which means that ΦP , as of the time of this work, cannot be computed without first computing P . Nevertheless, the SRM is used for a proof of concept of compressive deconvolution for VCD.

4.3 Compressive Deconvolution with Frequency Sub-Banding

Since the required size of Φ is linear in the sparsity of \mathbf{y} , if equation 4.1 could be broken down into several independent subproblems, each with a smaller measurement matrix than the one required for equation 4.1, then the previous compressive deconvolution algorithm could be sped up further. In this section, I propose a method to improve the compressive

¹See [20] for details of the ADMM algorithm.

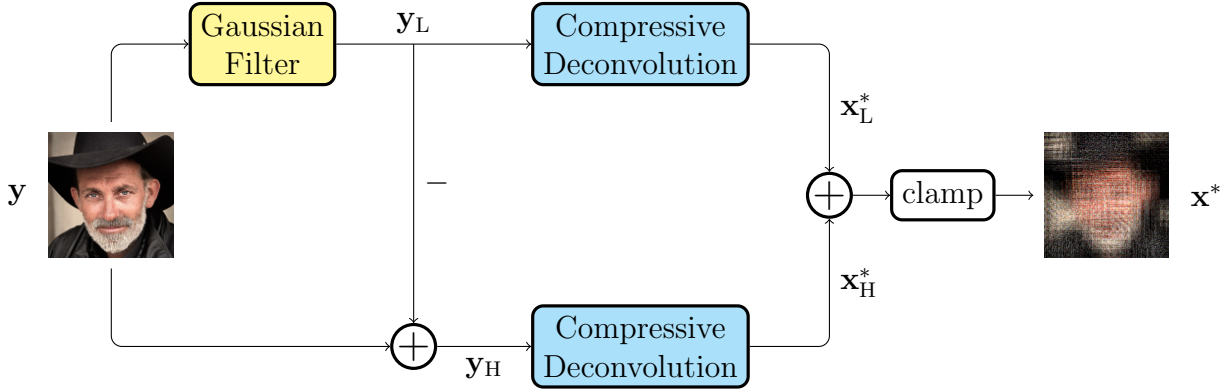


Figure 4.1: Compressive deconvolution with frequency sub-banding

deconvolution method in section 4.2 by splitting the target image into different frequency sub-bands before prefiltering. Since each sub-band should be sparser in the basis represented by Ψ , then compressive deconvolution should be able to perform better for each sub-band. Figure 4.1 illustrates this method.

Let \mathbf{y}_L be the result of applying a Gaussian filter to \mathbf{y} . Let $\mathbf{y}_H = \mathbf{y} - \mathbf{y}_L$. The prefiltered image \mathbf{x}^* is computed as

$$\mathbf{x}_L^* = \arg \min_{\mathbf{x}} \frac{1}{2\mu} \|\Phi \mathbf{y}_L - \Phi P \mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_p^p + \gamma \|\Psi^{-1} P \mathbf{x}\|_1 \quad \text{s.t. } 0 \leq \mathbf{x} \leq 1 \quad (4.2)$$

$$\mathbf{x}_H^* = \arg \min_{\mathbf{x}} \frac{1}{2\mu} \|\Phi \mathbf{y}_H - \Phi P \mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_p^p + \gamma \|\Psi^{-1} P \mathbf{x}\|_1 \quad \text{s.t. } 0 \leq \mathbf{x} \leq 1 \quad (4.3)$$

$$\mathbf{x}^* = \text{clamp}(\mathbf{x}_L^* + \mathbf{x}_H^*, 0, 1) \quad (4.4)$$

Since eq. (4.3) and eq. (4.4) can be solved independently of each other, they can be parallelized, resulting in a runtime that is similar to that of the existed compressive deconvolution method.

4.4 Evaluation

The proposed method was evaluated for the forward method with the parameters in table 4.1. These are the same parameters used by Parande in [20] (i.e., a 326 PPI display and a presbyopic eye under normal lighting conditions). However, there is one major difference. In [20], the sampling rate for the forward matrix was 1×1 samples per display pixel. However, as shown in section 2.3, that is equivalent to the Point-to-Point method. Therefore, in this work, a sampling rate of 5×5 samples per display pixel was used instead. Additionally, the sigma of the Gaussian filter was empirically chosen as 0.5.

The target images used are shown in figure 4.2. The bunny image shown in figure 4.2a is the same image that was used in the experiments in [20]. However, in this work, the

Parameter	Value
μ	1×10^{-2}
γ	1×10^{-3}
Focal length	20 mm
Aperture radius	1.5 mm
Near point distance	500 mm
Screen pixel pitch	0.078 mm
Pinhole mask separation distance	6 mm
Viewing distance	300 mm
Forward method sampling rate	5×5
Gaussian filter sigma	0.5

Table 4.1: Parameters for compressive deconvolution with frequency sub-banding experiments



(a) Bunny



(b) Cowboy

Figure 4.2: Target images for compressive deconvolution with frequency sub-banding experiments

same experiments are also run on the cowboy image shown in figure 4.2b, which contains more high-frequency signal than the bunny image. Together, these should provide a fair comparison with the existing compressive deconvolution method described in section 4.2.








Figure 4.3 shows the retinal projections of the prefiltered target images at different compressive sampling (CS) ratios (i.e., the fraction of measurements subsampled by Φ). Although subtle, the results with frequency sub-banding are slightly less grainy and noisy than the results without frequency sub-banding. This is further highlighted in figure 4.2, which shows the peak signal-to-noise ratio (PSNR) of both methods, and figure 4.3, which shows the structural similarity (SSIM) [25] of both methods. Additionally, figure 4.4 shows the perceptual loss, which has been shown to produce more visually pleasing results for tasks such as image super-resolution [16]. For an $n \times n$ target image \mathbf{y} , prefiltered display image

\mathbf{x}^* , and $\phi_j(\cdot)$, which denotes the output of layer j of a pretrained VGG-16 model with size $C \times m \times m$, the perceptual loss is given by

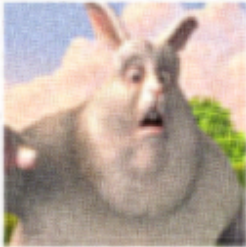


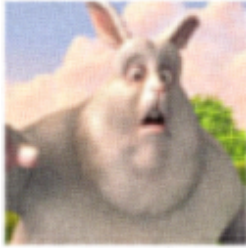

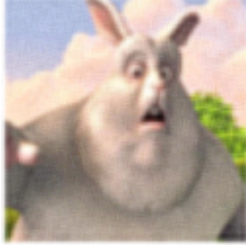
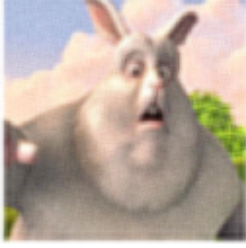
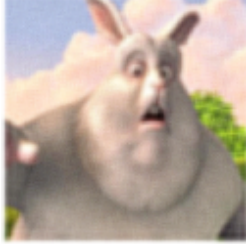
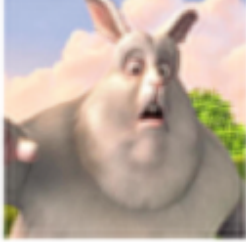

$$\mathcal{L}_{\phi_j}(\mathbf{y}, \mathbf{x}^*) = \frac{1}{Cm^2} \|\phi_j(\mathbf{y}) - \phi_j(P\mathbf{x}^*)\|_2^2 \quad (4.5)$$

For the perceptual loss, unlike PSNR and SSIM, lower values represent higher image quality. In figure 4.4, the outputs of layer `relu2_2` are used as the features for perceptual loss.











For all three metrics, the method with frequency sub-banding is consistently able to outperform the method without frequency sub-banding. More specifically, for most CS ratios (from about 0.2 to about 0.8), the visual quality of the method with frequency sub-banding is on-par with the visual quality of the method without frequency sub-banding, but with a 10% higher CS ratio. Only at CS ratios greater than 0.9 does the existing compressive deconvolution method outperform the proposed method. This suggests two things: (1) at most CS ratios, the visual quality can be improved by proposed method, and (2) the proposed method can offer an improvement in runtime over the existing compressive deconvolution method by simply reducing the CS ratio while maintaining visual quality.

CS Ratio	Without Sub-Banding	With Sub-Banding
0.1		
0.2		
0.3		
0.4		
0.5		











(a) Bunny

CS Ratio	Without Sub-Banding	With Sub-Banding
0.6		
0.7		
0.8		
0.9		
1.0		

(a) Bunny (cont.)

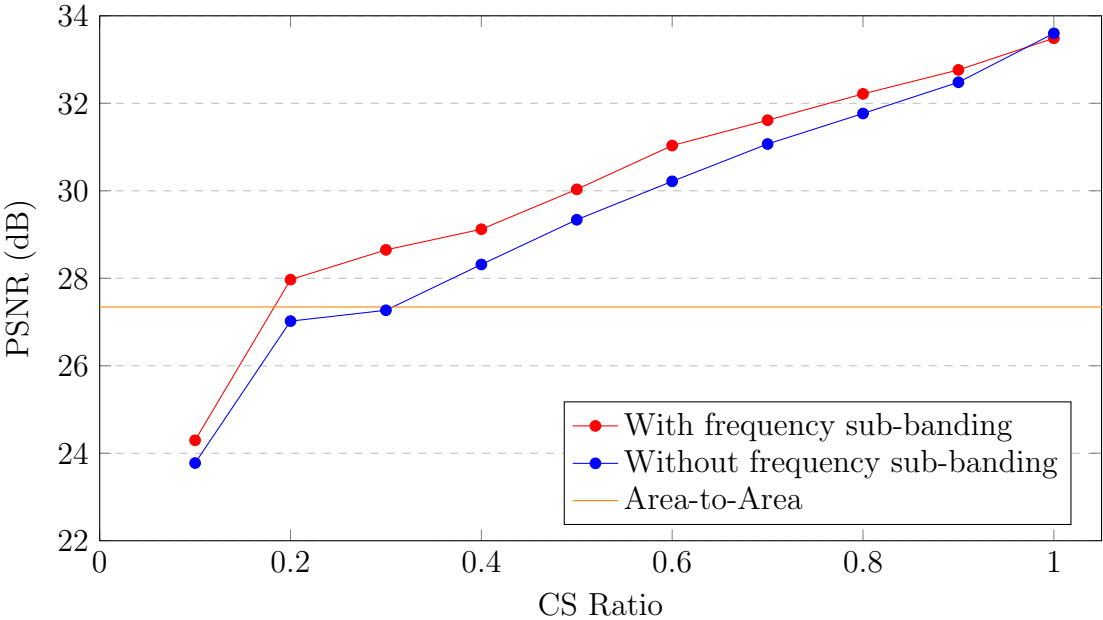
CS Ratio	Without Sub-Banding	With Sub-Banding
0.1		
0.2		
0.3		
0.4		
0.5		

(b) Cowboy

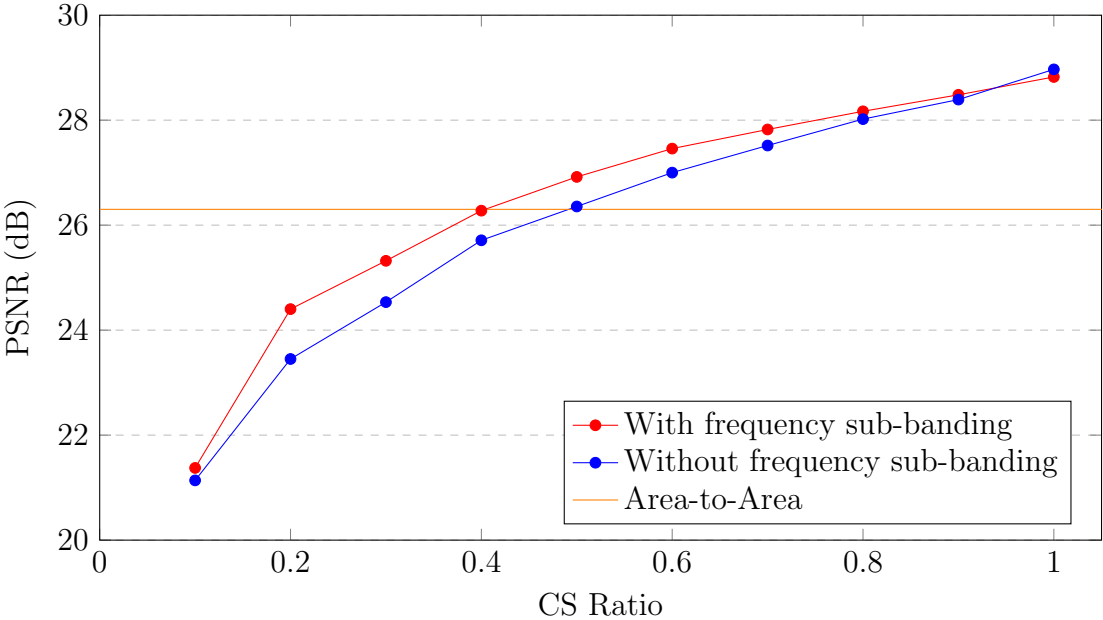
CS Ratio	Without Sub-Banding	With Sub-Banding
0.6		
0.7		
0.8		
0.9		
1.0		

(b) Cowboy (cont.)

Figure 4.3: Retinal projections of compressive deconvolution prefilters with and without frequency sub-banding

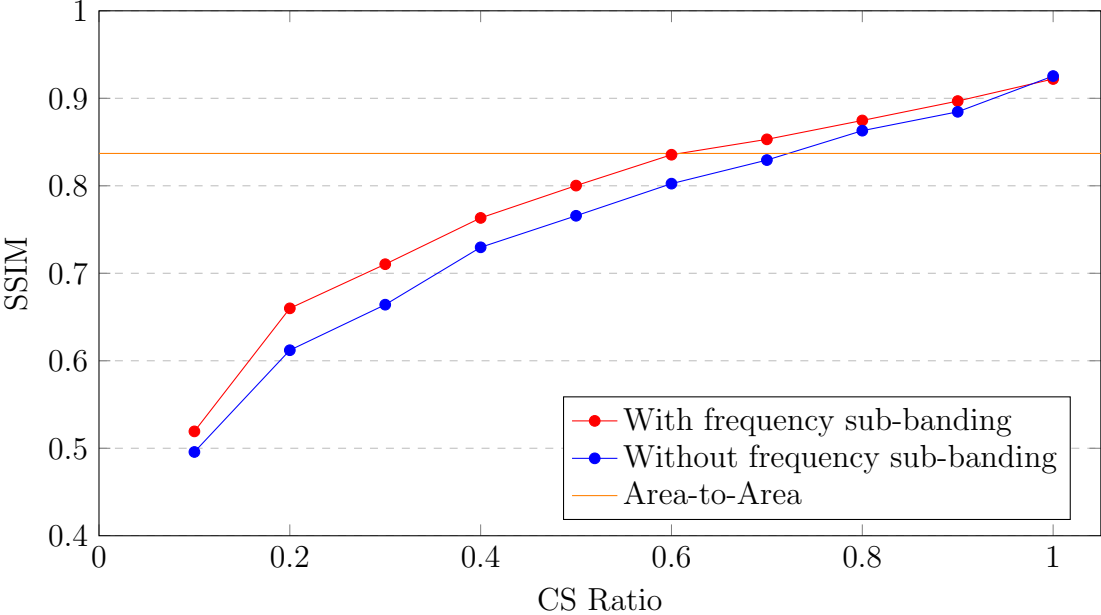


(a) Bunny

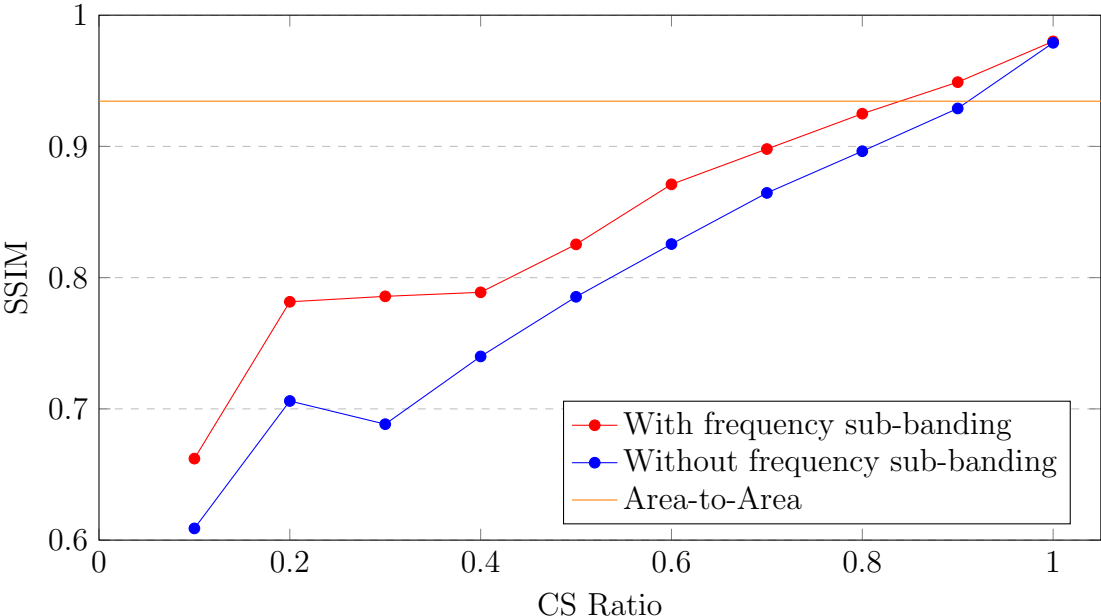


(b) Cowboy

Figure 4.2: Comparison of PSNR for compressive deconvolution with and without frequency sub-banding

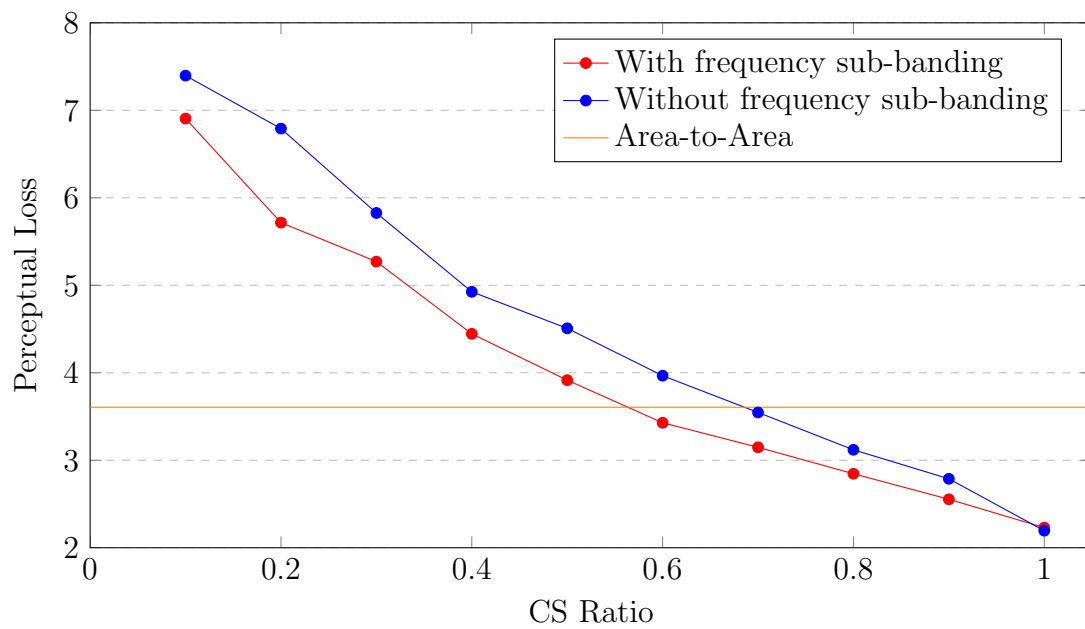


(a) Cowboy

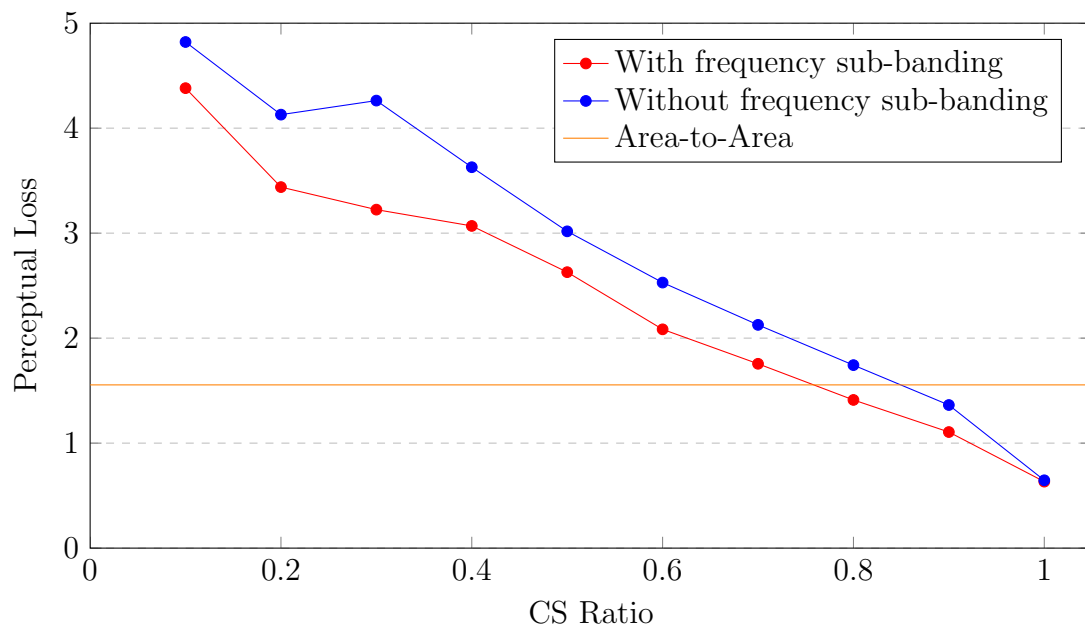


(b) Bunny

Figure 4.3: Comparison of SSIM for compressive deconvolution with and without frequency sub-banding



(a) Cowboy



(b) Bunny

Figure 4.4: Comparison of perceptual loss (VGG-16 layer relu2.2) for compressive deconvolution with and without frequency sub-banding

Chapter 5

Deconvolution with Vision Transformers

5.1 Motivation

Optimization-based compressive sampling is restricted by the sparsity assumption and the slow runtime of optimization procedures required for reconstruction. In recent years, deep-learning (DL) approaches to compressive sampling have been demonstrated to outperform optimization-based approaches [19]. In fact, optimization-based compressive sampling can be thought of as an autoencoder, where the encoder is given by the measurement matrix Φ , and the decoder is given by the optimization routine that solves eq. (3.5).

DL methods have also been shown to outperform classical techniques for image deblurring [28]. Assuming that the blurring function b is known, the goal of image deblurring is to recover a sharp image \mathbf{y} given a blurred image $\mathbf{x} = b(\mathbf{y})$. Although b is typically considered to be some combination of motion blur, out-of-focus blur, and Gaussian blur, we can also consider $b(\mathbf{y}) = P\mathbf{y}$. Thus, we would like to find a model f such that for a target image \mathbf{x} , $\mathbf{x} = Pf(\mathbf{x})$. Of course, $f(\mathbf{x}) = P^T (PP^T)^{-1} \mathbf{x}$ is one possible model. However, this model would be so large that it would not fit into memory on most devices (for a 128×128 image and 5×5 display pixels per macropixel, the model would have over 6.7 billion parameters and require nearly 27 GB of memory). Thus, we would like to find a smaller model that could realistically run on most devices in real-time (at least 30 frames per second).

5.2 Methodology

Model Overview

Historically, convolutional neural networks (CNNs) have been a popular neural network architecture for image processing tasks. However, Vision Transformers (ViTs), first introduced by Dosovitskiy et al. in [11], offer several advantages over CNNs for the VCD task. Based

on the Transformer architecture proposed by Vaswani et al. in [23], Vision Transformers process images not as a grid of pixels, but rather as a sequence of tokens, similar to how sentences are processed in natural language processing tasks. Each input image is split into non-overlapping patches, and each patch is linearly projected into a high-dimensional space. These projected patches serve as the tokens into the Transformer. The self-attention mechanism combined with positional encodings allows similarity scores to be computed between every pair of tokens based on their content and position within the image. It is this self-attention mechanism that provides the advantages over CNNs. First, in order for each neuron to achieve a large receptive field, CNNs require a large number of layers. On the other hand, in every self-attention layer of a ViT, each token is able to attend to every other token. This means that even shallow ViTs have a large receptive field. Second, because CNNs are based on convolutions, which are spatially-invariant, CNNs have a harder time learning to prefilter different parts of an image based on their position within the image. However, by adding a positional encoding to the input tokens, ViTs do not have the same inductive bias of spatial-invariance. This is important because the way that a display pixel projects onto the retina depends on the position of that pixel.

For these reasons, I propose a ViT-based model for the VCD task, illustrated in figure 5.1. First, the target image \mathbf{x} is split into non-overlapping patches of size $p \times p$ and flattened into p^2 -dimensional vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_\ell\}$. Next, each patch is linearly projected into a d -dimensional space:

$$\mathbf{v}_i = W_1 \mathbf{x}_i + \mathbf{b}_1 \quad (5.1)$$

This sequence of projected patches is then fed into a Transformer encoder:

$$\mathbf{z}_i = E(\mathbf{v}_i) \quad (5.2)$$

Finally, the outputs of the encoder are linearly projected into $u^2 p^2$ -dimensional vectors with a sigmoid function to restrict output values to be between 0 and 1:

$$\mathbf{y}_i = \sigma(W_2 \mathbf{z}_i + \mathbf{b}_2) \quad (5.3)$$

The vectors $\{\mathbf{y}_1, \dots, \mathbf{y}_\ell\}$ are then reshaped into patches of size $up \times up$ and combined back together into the final prefiltered image \mathbf{y} .

Spatially Separable Self-Attention

In standard self-attention, attention scores are computed for every pair of tokens, resulting in an $\mathcal{O}(\ell^2)$ runtime. For large images and/or small patch sizes, this can be impractical. Thus, other types of attention mechanisms have been proposed that take advantage of the spatial structure of images. One such mechanism is *spatially separable self-attention* (SSSA), which is used in the Twins-SVT model architecture [7]. SSSA is composed of *locally-grouped self-attention* (LSA) and *global sub-sampled attention* (GSA).

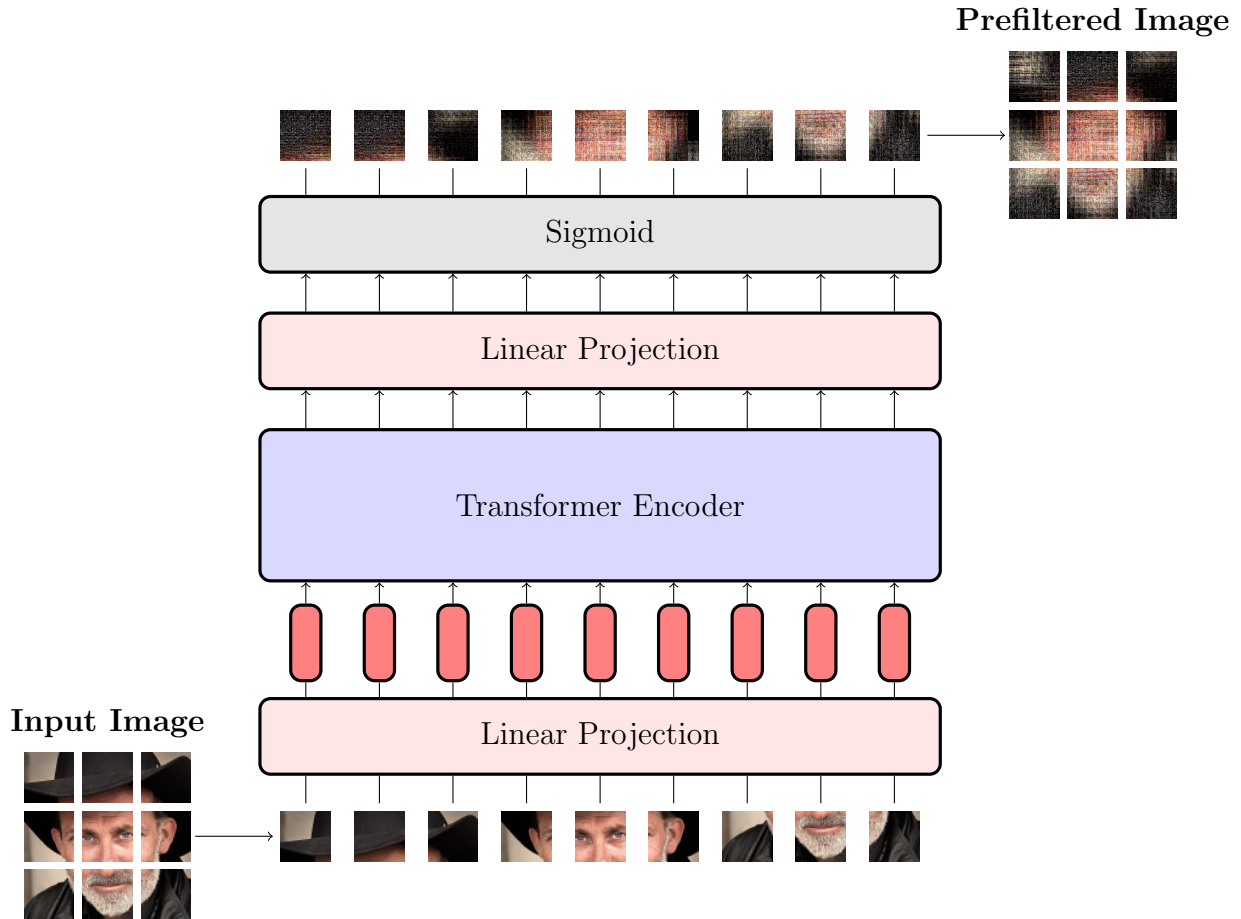


Figure 5.1: Overview of Vision Transformer-based prefiltering. Although the input and prefiltered image are shown in color, for a multi-channel color image, each channel is prefiltered separately, similar to compressive deconvolution.

Locally-Grouped Self-Attention (LSA)

For an $n \times n$ image, patch-size $p \times p$, and embedding dimension d , the input into each Transformer encoder layer can be thought of as an image of size $\frac{n}{p} \times \frac{n}{p} \times d$. This image can then be split into non-overlapping windows of size $s \times s$. Self-attention is then computed only within elements of the same window. For each window, the runtime of computing self-attention is $\mathcal{O}(s^4)$. Since there are $\frac{n}{ps} \times \frac{n}{ps}$ total windows, the total runtime of computing LSA for an image is $\mathcal{O}\left(\frac{s^2 n^2}{p^2}\right)$. Since the runtime for standard self-attention is $\mathcal{O}\left(\frac{n^4}{p^4}\right)$, then if $s < \frac{n}{p}$, the runtime complexity of LSA is faster than standard self-attention.

Global Sub-Sampled Attention (GSA)

In LSA, each element can only directly attend to elements in the same window. Thus, GSA aims to improve cross-window information exchange. First, each window is summarized into a single vector via a strided convolution of kernel size $s \times s$ and stride s . Attention is then computed, where the elements of the input into the GSA layer serve as the queries, and the summary vectors serve as the keys and values. The runtime of GSA is $\mathcal{O}\left(\frac{n^4}{p^4 s^2}\right)$, compared to $\mathcal{O}\left(\frac{n^4}{p^4}\right)$ for standard self-attention. Together with LSA, the total runtime is $\mathcal{O}\left(\frac{s^2 n^2}{p^2} + \frac{n^4}{p^4 s^2}\right)$. Thus, if $\frac{s^4}{s^2-1} < \frac{n^2}{p^2}$, LSA and GSA together have lower runtime complexity than standard self-attention.

Transformer Encoder Architecture

Each layer in the proposed Transformer encoder with SSSA is shown in figure 5.2. In a standard Transformer, the positional encoding is added to the input sequence only once before being passed into the Transformer. However, inspired by Twins-PCPVT [7], positional encoding is added before each encoder layer. This is because in the VCD problem, the position of each pixel is significantly more important than in most other image processing tasks. In fact, if the model is a perfect psuedoinverse of P , then only the position of a pixel in the target image affects how that pixel is prefiltered. Therefore, in the proposed model, the positional information is re-added multiple times throughout the model. However, unlike Twins-PCPVT, rather than a conditional positional encoding [6], a standard sinusoidal positional encoding is used instead.

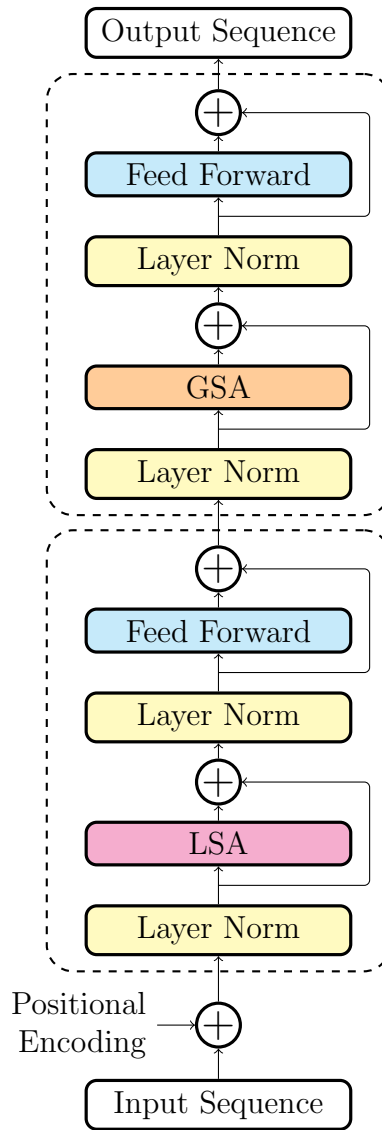


Figure 5.2: Transformer encoder layer using *spatially separable self-attention* (SSSA)

5.3 Experiments

Loss Functions

Since the choice of loss function can greatly impact the visual quality of retinal projection of an image prefilter, multiple models were trained, each with a different loss function. For an $n \times n$ target image \mathbf{x} and a prefiltered display image $\hat{\mathbf{y}}$, the mean squared error (MSE)

and mean absolute error (MAE) losses are given by

$$\mathcal{L}_{\text{MSE}}(\mathbf{x}, \hat{\mathbf{y}}) = \frac{1}{n^2} \|\mathbf{x} - P\hat{\mathbf{y}}\|_2^2 \quad (5.4)$$

$$\mathcal{L}_{\text{MAE}}(\mathbf{x}, \hat{\mathbf{y}}) = \frac{1}{n^2} \|\mathbf{x} - P\hat{\mathbf{y}}\|_1 \quad (5.5)$$

The VGG-16 `relu1_1` perceptual loss is given by

$$\mathcal{L}_{\phi_1}(\mathbf{x}, \hat{\mathbf{y}}) = \frac{1}{Cm^2} \|\phi_1(\mathbf{x}) - \phi_1(P\hat{\mathbf{y}})\|_2^2 \quad (5.6)$$

which follows the same formulation as eq. (4.5).

Similar to Swin2SR [8], a Transformer-based image super-resolution model, an additional high-frequency loss term is added to the MAE loss:

$$\mathcal{L}_{\text{MAE+HF}}(\mathbf{x}, \hat{\mathbf{y}}) = \frac{1}{n^2} \|\mathbf{x} - P\hat{\mathbf{y}}\|_1 + \frac{1}{n^2} \|HF(\mathbf{x}) - HF(P\hat{\mathbf{y}})\|_1 \quad (5.7)$$

$HF(\mathbf{i})$ denotes the high-frequency information of image \mathbf{i} and is computed with a 5×5 Gaussian filter g as

$$HF(\mathbf{i}) = \mathbf{i} - \mathbf{i} * g \quad (5.8)$$

Since $HF(\cdot)$ only computes activations for one filter whereas VGG-16 computes activations for multiple filters, the perceptual loss is also added to the MAE loss:

$$\mathcal{L}_{\text{MAE}+\phi_1}(\mathbf{x}, \hat{\mathbf{y}}) = \mathcal{L}_{\text{MAE}}(\mathbf{x}, \hat{\mathbf{y}}) + \mathcal{L}_{\phi_1}(\mathbf{x}, \hat{\mathbf{y}}) \quad (5.9)$$

Since the VGG-16 `relu1_1` perceptual loss is typically about 10 times the MAE loss, the perceptual loss can be weighted by 1/10 to keep the two loss components roughly equal:

$$\mathcal{L}_{\text{MAE}+\phi_1, \text{weighted}}(\mathbf{x}, \hat{\mathbf{y}}) = \mathcal{L}_{\text{MAE}}(\mathbf{x}, \hat{\mathbf{y}}) + \frac{1}{10} \mathcal{L}_{\phi_1}(\mathbf{x}, \hat{\mathbf{y}}) \quad (5.10)$$

Model Details

Each model uses the same configuration (determined empirically), which is detailed in table 5.1. The display, eye, and forward method sampling rate are the same as in section 4.4. Table 5.2 also shows the runtime of the model per 3-channel color image, along with the ray-tracing methods and forward method for comparison. The runtime is measured on a 2020 M1 MacBook Air, and images were not batched together when measuring runtime.

Training Details

Each model was trained on the Linnaeus 5 dataset [5], which consists of 5 classes of images: berry, bird, dog, flower, other. Each image has size 128×128 , and each class contains

Number of layers	6
Number of attention heads	8
Image patch size	4×4
SSSA window size	16×16
Embedding dimension	64
Number of parameters	6.9M

Table 5.1: Transformer model configuration details

Method	Time (s)
Point-to-Point	0.014
Many-to-Many	1.266
Area-to-Area	0.030
Forward Method	163.49
Transformer (w/o GPU)	0.098
Transformer (w/ GPU)	0.058

Table 5.2: Runtime (per image) comparison of ray-tracing methods, forward method, and Transformer model.

1,200 training images and 400 test images. Additionally, each training set is split into an 80%/20% training-validation split. The images are converted to single-channel grayscale images for training.

Each batch consists of 32 images, and each model was trained for 1,000 epochs. The AdamW optimizer [18] was used with a learning rate of 1×10^{-3} .

Results

Table 5.3 shows the average test PSNR, SSIM, and VGG-16 `relu2_2` perceptual loss for each model. Across all three metrics, the model trained with the weighted MAE + VGG-16 `relu1_1` perceptual loss performed the best. Figure 5.3 shows the retinal projections for the bunny and cowboy images using the model trained with the weighted MAE + VGG-16 `relu1_1` perceptual loss. The Point-to-Point, Many-to-Many, and Area-to-Area methods are also shown as a baseline, and the quantitative metrics are shown in table 5.4. Not only do the quantitative metrics show an improvement in quality, visually, the results with the Transformer model appear to contain more of the high-frequency content of the original image.

Trained loss function	PSNR (dB)	SSIM	Perceptual loss (relu2_2)
MSE	26.26	0.82	3.42
MAE	26.38	0.84	3.19
Perceptual loss (ϕ_1)	26.20	0.84	2.95
MAE+HF	26.52	0.85	3.02
MAE+ ϕ_1 (unweighted)	26.67	0.85	2.98
MAE+ ϕ_1 (weighted)	26.93	0.86	2.89

Table 5.3: Average test PSNR, SSIM, and perceptual loss (VGG-16 layer relu2_2) by loss function used for training



Figure 5.3: Visual comparison of Point-to-Point, Many-to-Many, Area-to-Area, and Transformer methods. The Transformer method shown was trained with MAE + relu1_1 VGG-16 perceptual loss (weighted).

Trained loss function	PSNR (dB)	SSIM	Perceptual loss (relu2_2)
Point-to-Point	23.04	0.86	2.52
Many-to-Many	24.36	0.91	1.71
Area-to-Area	27.59	0.93	1.55
MSE	29.54	0.93	1.63
MAE	29.85	0.94	1.38
Perceptual loss (ϕ_1)	29.48	0.93	1.47
MAE+HF	29.82	0.95	1.29
MAE+ ϕ_1 (unweighted)	29.96	0.93	1.44
MAE+ ϕ_1 (weighted)	30.51	0.95	1.27

(a) Bunny

Trained loss function	PSNR (dB)	SSIM	Perceptual loss (relu2_2)
Point-to-Point	23.37	0.73	4.57
Many-to-Many	25.37	0.83	3.48
Area-to-Area	26.30	0.84	3.61
MSE	26.36	0.83	3.75
MAE	26.44	0.85	3.47
Perceptual loss (ϕ_1)	25.98	0.83	3.27
MAE+HF	26.53	0.86	3.35
MAE+ ϕ_1 (unweighted)	26.53	0.85	3.36
MAE+ ϕ_1 (weighted)	26.89	0.87	3.20

(b) Cowboy

Table 5.4: Comparison of PSNR, SSIM, and perceptual loss (VGG-16 layer relu2_2) for bunny and cowboy images by loss function used for training. Ray-tracing methods included as a baseline.

Chapter 6

Future Work

This work provides an improvement to the existing optimization-based compressive deconvolution algorithm proposed by Parande in [20], and introduces a non-iterative Vision Transformer-based model that improves upon the visual quality of previous ray-tracing methods while still being faster than previous optimization-based algorithms. However, in order for the proposed methods to be useful, they must be able to work in real-time, scale to larger images, correct different aberrations.

6.1 Faster Compressive Deconvolution

In order for compressive deconvolution to run in real-time, several improvements should be made. First, the optimization routine used should be faster, either through better implementation or by switching to a different optimization routine altogether. Second, the measurement matrix Φ should be one where ΦP can be constructed together, thus eliminating the need for the entire P matrix to be constructed. However, for ΦP to be useful, it should also be sparse in order to be able to fit into memory on most devices. Currently, since P is sparse and Φ is implemented using linear operators, this is not an issue, but in the future, when Φ is not the SRM, this will be important.

6.2 Improving Vision Transformers

Although Vision Transformers offer better visual quality than ray-tracing methods and better speed than optimization-based methods, the model proposed in this work is less flexible than other methods. First, a single model can only prefilter an image during inference-time if all the parameters (device, visual aberration, viewing distance, image size, etc.) are the same as those during training-time. Thus, in order to improve the flexibility of a model, at the very least, it should be designed and trained such that it can adapt to different viewing distances, since users are likely to move around while they view the display.

Additionally, the model proposed in this work is unable to prefilter a video stream in real time without batching, and is only able to prefilter images of size 128×128 . Future work should be done to further improve the speed of the model, and further work should investigate the ability of the model to scale to larger image sizes.

6.3 Higher-Order Aberrations

This work only focuses on far-sightedness, which is a lower-order aberration. However, a significant advantage of vision-correction displays over external eyewear is the ability to correct higher-order aberrations as well. In theory, as long as P models a user's visual aberrations, the methods proposed in this work should still be applicable, but further work should investigate any potential limitations or improvements to these methods for higher-order aberrations.

Chapter 7

Conclusion

Previous approaches to the VCD problem could be categorized into two categories: ray-tracing techniques and optimization-based techniques. While ray-tracing techniques are fast and highly parallelizable, optimization-based techniques provide better visual quality. Compressive deconvolution is a method of bridging the gap between the speed of ray-tracing techniques and accuracy of optimization-based techniques. This work proposes a method of further improving compressive deconvolution, offering improved visual quality and a potential runtime improvement. This could help further improve optimization-based techniques and bring their runtime closer to that of ray-tracing techniques. With further improvement, optimization-based techniques could potentially be run in real-time, which is important for VCD to be applicable to the real world.

This work also introduces a third category of VCD approaches: deep-learning techniques. The deep-learning method proposed in the work is based on Vision Transformers with spatially-separable self-attention. However, this is only one possible approach, which this work has shown to be faster than optimization-based techniques and more accurate than ray-tracing techniques. Deep-learning techniques for VCD need not be limited to this approach. In general, deep-learning techniques are faster than optimization-based techniques since there is no inherent need for iteration during inference-time. They also have the benefit of being able to learn the structure of natural images by training with large amounts of data, helping them improve upon the visual quality of ray-tracing techniques. Additionally, unlike compressive deconvolution, the learned natural image structure is inherently coupled with the prefiltering process, whereas compressive deconvolution separates the image measurement and prefilter projection matrices. With future work, deep-learning approaches are likely to become fast enough for real-time usage with better accuracy than ray-tracing methods. By demonstrating the applicability of deep-learning to the VCD problem, hopefully the gap between ray-tracing techniques and optimization-based techniques can be crossed, making vision-correcting displays even more practical and helpful for the millions of people with visual aberrations.

Bibliography

- [1] Richard H Byrd et al. “A limited memory algorithm for bound constrained optimization”. In: *SIAM Journal on scientific computing* 16.5 (1995), pages 1190–1208.
- [2] Emmanuel J. Candes and Michael B. Wakin. “An Introduction To Compressive Sampling”. In: *IEEE Signal Processing Magazine* 25.2 (2008), pages 21–30. DOI: 10.1109/MSP.2007.914731.
- [3] Emmanuel Candès and Justin Romberg. “Sparsity and incoherence in compressive sampling”. In: *Inverse Problems* 23.3 (Apr. 2007), pages 969–985. ISSN: 1361-6420. DOI: 10.1088/0266-5611/23/3/008. URL: <http://dx.doi.org/10.1088/0266-5611/23/3/008>.
- [4] Emmanuel J. Candès. “The restricted isometry property and its implications for compressed sensing”. In: *Comptes Rendus Mathématique* 346.9 (2008), pages 589–592. ISSN: 1631-073X. DOI: <https://doi.org/10.1016/j.crma.2008.03.014>. URL: <https://www.sciencedirect.com/science/article/pii/S1631073X08000964>.
- [5] G Chaladze and L Kalatozishvili. “Linnaeus 5 dataset for machine learning”. In: *chaladze.com* (2017).
- [6] Xiangxiang Chu et al. *Conditional Positional Encodings for Vision Transformers*. 2023. arXiv: 2102.10882 [cs.CV].
- [7] Xiangxiang Chu et al. *Twins: Revisiting the Design of Spatial Attention in Vision Transformers*. 2021. arXiv: 2104.13840 [cs.CV].
- [8] Marcos V. Conde et al. *Swin2SR: SwinV2 Transformer for Compressed Image Super-Resolution and Restoration*. 2022. arXiv: 2209.11345 [cs.CV].
- [9] The Vision Council. *Organizational Overview*. 2021. URL: https://thevisioncouncil.org/sites/default/files/assets/media/TVC_OrgOverview_sheet_2021.pdf (visited on 05/12/2024).
- [10] Thong T. Do et al. “Fast and Efficient Compressive Sensing using Structurally Random Matrices”. In: *CoRR* abs/1106.5037 (2011). arXiv: 1106.5037. URL: <http://arxiv.org/abs/1106.5037>.
- [11] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

- [12] Stanford Medicine Children’s Health. *Refractive Errors in Children*. URL: <https://www.stanfordchildrens.org/en/topic/default?id=refractive-errors-in-children-90-P02098> (visited on 05/12/2024).
- [13] Fu-Chung Huang. “A Computational Light Field Display for Correcting Visual Aberrations”. PhD thesis. EECS Department, University of California, Berkeley, Dec. 2013. URL: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-206.html>.
- [14] Fu-Chung Huang et al. “Correcting for optical aberrations using multilayer displays”. In: *ACM transactions on graphics (TOG)* 31.6 (2012), pages 1–12.
- [15] Fu-Chung Huang et al. “Eyeglasses-free display: towards correcting visual aberrations with computational light field displays”. In: *ACM Trans. Graph.* 33.4 (July 2014). ISSN: 0730-0301. DOI: 10.1145/2601097.2601122. URL: <https://doi.org/10.1145/2601097.2601122>.
- [16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*. 2016. arXiv: 1603.08155 [cs.CV].
- [17] Michael Land. “Focusing by shape change in the lens of the eye: a commentary on Young (1801) ‘On the mechanism of the eye’”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 370.1666 (Apr. 2015), page 20140308. ISSN: 1471-2970. DOI: 10.1098/rstb.2014.0308. URL: <http://dx.doi.org/10.1098/rstb.2014.0308>.
- [18] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].
- [19] Alina L. Machidon and Veljko Pejovic. *Deep Learning Techniques for Compressive Sensing-Based Reconstruction and Inference – A Ubiquitous Systems Perspective*. 2021. arXiv: 2105.13191 [eess.SP].
- [20] Anmol Parande. “Compressive Deconvolution Algorithms for a Computational Light-field Display for Correcting Visual Aberrations”. Master’s thesis. EECS Department, University of California, Berkeley, May 2022. URL: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-110.html>.
- [21] C.E. Shannon. “Communication in the Presence of Noise”. In: *Proceedings of the IRE* 37.1 (1949), pages 10–21. DOI: 10.1109/JRPROC.1949.232969.
- [22] Mittanamalli S Sridhar. “Anatomy of cornea and ocular surface”. In: *Indian Journal of Ophthalmology* 66.2 (2018), pages 190–194.
- [23] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].
- [24] Madeleine Vessel. *Higher-order aberrations*. URL: <https://www.allaboutvision.com/conditions/aberrations.htm> (visited on 05/12/2024).

- [25] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pages 600–612. DOI: 10.1109/TIP.2003.819861.
- [26] Zehao Wu. “Investigating Computational Approaches and Proposing Hardware Improvement to the Vision Correcting Display”. Master’s thesis. EECS Department, University of California, Berkeley, May 2016. URL: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-67.html>.
- [27] Shichao Yue. *Introduction to Vision Correcting Display*. Technical report. University of California, Berkeley, 2015.
- [28] Kaihao Zhang et al. *Deep Image Deblurring: A Survey*. 2022. arXiv: 2201.10700 [cs.CV].
- [29] Yirong Zhen. “New algorithms for the vision correcting display”. Master’s thesis. EECS Department, University of California, Berkeley, 2019.