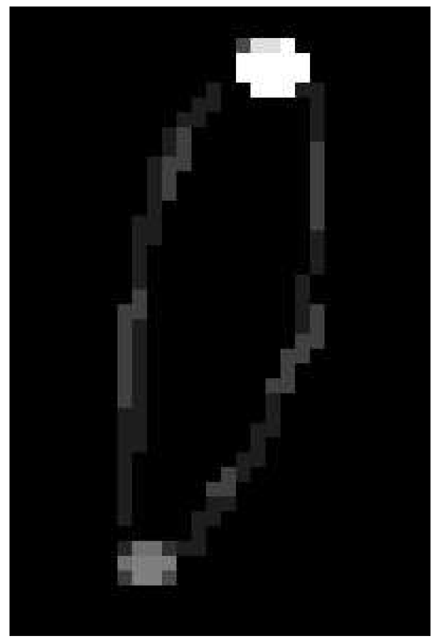
For my project, I'd like to do a reverse-convolution type of project involving a medical condition of mine. I have keratoconus in my right eye, which is a condition where the lens of the eye deforms. The lens's primary job is to focus light through the pupil onto the retina (exactly analogously to a lens on a conventional camera), and its deformation causes different parts of the lens to focus light at slightly offset angles when compared to a properly shaped lens as a function of the part of the lens the light travels through (roughly independent to the incidence angle of the light for small neighborhoods). The deformation is very small in magnitude - normal lenses are typically consistently $\sim 450\mu m$ thick, but mine vary from $380\mu m$ to $440\mu m$ or so. This, combined with the fact that humans typically only actively focus on a small neighborhood at the center of their vision, mean that the effect on what I perceive through my right eye is nearly exactly the ground truth convolved with a relatively small kernel. The deformation is also fairly smooth, making this kernel relatively sparse (this is both true for me and in general - the most common form of keratoconus is caused by part of the lens "drooping down" - this results in a high weight where the unaffected parts of the vision focus and a high weight where the main part of the deformation focuses with small weights corresponding to areas on the boundary of the deformation and zero elsewhere - I’d estimate my kernel to be similar to the image on the right where a normal person’s vision would just be the white part).



Since I have other (unrelated but significantly worse) issues in my left eye, I often end up primarily relying on my right eye to use the computer. This doesn’t typically cause issues for me with the exception of resolving low-contrast dark detail that occurs directly below brighter, high-contrast objects (the reason why should be obvious from the kernel haha)

What I’d like to do for my project is to make an interactive application that lets the user find their keratoconus kernel then make a solver that, for a given image, finds an image that, when convolved with this kernel, yields an image close to the original. Note that this is similar to deconvolving motion blurs but is slightly different in that there may not exist a prior that properly solves the problem for typical distance metrics. For instance, the prior for a single white dot on a black background would need to have a negative radiance below the white dot which is not possible. That being said, it should still be possible to find an image where the proportional difference between spatially close pixels is close to that in original (for example, 0.9*orig_image+0.1 would be ‘very close’ in this context while being quite a poor result from a conventional distance perspective) due to the ability to effectively get negative radiance in areas by making all blacks lighter.

The primary part of my project will be finding a good loss function for this ‘contrast-oriented perceptual closeness’ and making a solver to find priors that give good results with it. I have found some research to guide my initial steps into this problem. Firstly, “Image Quality

Assessment: From Error Visibility to Structural Similarity” by Zhou Wang et. al. explores how to objectively evaluate image error with regards to the human visual system. It focuses on how our perception is highly geared towards extracting structured information and provides an outline for evaluating the degradation of these visible structures. Next, since we do not have much information as to what the priors of this problem should look like, I feel that the “Unrolled Optimization with Deep Priors” paper introduced in class will be useful. The problem I am focusing on has a weird aspect of knowing the final image exactly and not having any initial distribution over what could generate it, so having a framework to learn the entire pipeline from a good prior should be very useful. Finally, I want to take some ideas from existing non-learned deconvolution techniques such as those explored in “Fast iteratively reweighted least squares for lp regularized image deconvolution and reconstruction” by Xu Zhou et. al. which looks at methods of deconvolution with a regularization function ||Rx||_p where R is a matrix. I feel as though solving the linear system isn’t directly useful due to the gap between the perceptual closeness I am looking to optimize for and conventional distance metrics, but I want to look for ways to encode some form of retaining structural closeness as a matrix for use with these more conventional regularized solvers.