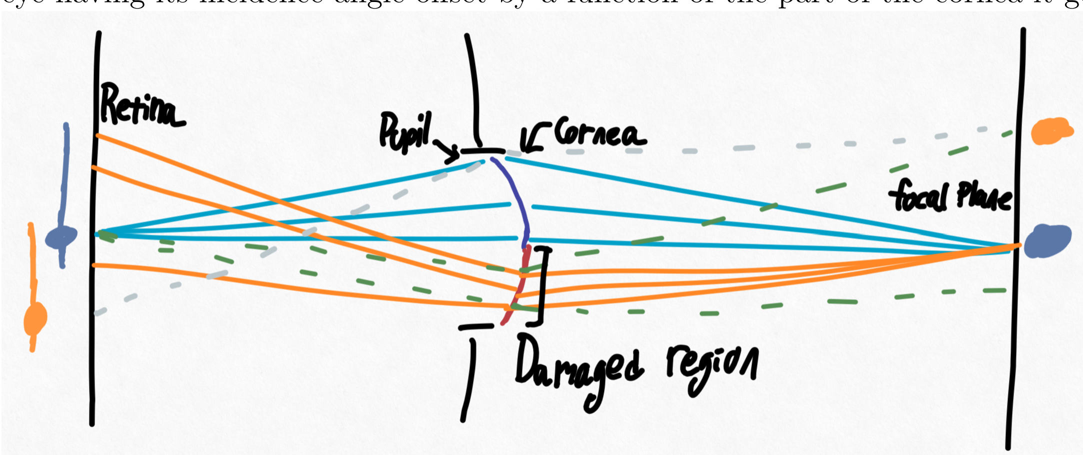
Inverting Keratoconus Refraction Errors

Paul Duggan (paulzd)

In this paper, we seek to create a methodology for displaying images on conventional screens that appear as sharp images to viewers with refraction errors from keratoconus. We outline a two-step process where we first use an interactive loop to learn the individual's refractive errors in the form of a 2d convolution kernel then minimize a structure-prioritizing error to find the image that, when convolved with this kernel, produces an image visually similar to the original. We find that the process of learning an individual's kernel is slow and cumbersome, but once an accurate model for their kernel is obtained, we can create a sharp image at the cost of locally brightening dark regions of the image.

Introduction

Keratoconus is a degenerative eye condition that results from the cornea, the part of the eye in front of the lens, becoming malformed over time. This results in all light entering the eye having its incidence angle offset by a function of the part of the cornea it goes through.



Above is a illustration showing how light passing through the damaged portion of the cornea results in a smeared image being projected onto the retina. One can see that the light from the purple dot entering the eye through the undamaged portion of the cornea gets focused to a single point on the retina. Rays entering through the damaged portion, however, are sent to other areas. Since the deflection angle offset induced by any point on the damaged region is fairly constant for similar angles of incident light, the image perceived by the eye is very close to the ground truth image convolved some PSF we denote A. Since the cornea remains relatively stable on a short term basis, this kernel is typically fairly static and can be formalized as $A = (1 - \alpha)K_g + \alpha K_b$ where K_g is the PSF of a healthy eye and K_b is the PSF of the damaged portion. We assume α isn't excessively close to 1 in this project (< 0.7 or so). This is not an unreasonable assumption to make; for most patients, keratoconus most strongly affects the lobes directly above and below the center of the pupil, leaving the sides fairly close to an undamaged cornea.

Here is an image of what the world might look like to someone with Keratoconus. Notice how the bright smears from the image can obscure dark, low contrast, structural detail below them (such as the detail on the horizon beneath the lamp poles). Our aim is to create a method for making these details more visible to people with keratoconus. If we have a model for the refractive error of an individual's retina, it should be possible to use gradient descent on error metrics in line with human perception to to make these lost structural details more apparent. This motivates our two-step process in which we first build an accurate estimate of a given user's PSF A then use a custom-tuned MSSIM error that greatly prioritizes structural detail.

Related Work

There currently exist no methods to cure or reverse keratoconus medically, barring risky and invasive surgeries such as cornea transplants. The most common way to restore vision quality is to physically correct the shape of the cornea by using a hard contact lens with a similar IOR to the cornea to mitigate refraction disparities caused by its uneven thickening and thinning. However, these contacts must be machined to fit the precise shape of the user's cornea. Furthermore, every application and removal has the risk of causing more cornea damage. Most limitingly, wearing these contacts for long periods of time is both uncomfortable and unhealthy for the eye.

Aside from direct corrections to the eye's biomechanics, there is research on modifying the image source to correct general refractive errors. Lightfields can create an image plane where both the position and direction of outgoing light rays are under control. Through this, the part of the pupil a given pixel's ray goes through can be chosen, allowing for the direct accommodation of the PSF and for the desired image to be created on the retina. This method is very sensitive to the pupil position in all three axis of space as well as its rotation, and requires specialized hardware to be used. Furthermore, there is a tradeoff between the spatial and angular resolution of these lightfields, meaning that for people with very unsmooth PSF kernels, for the technique to be effective, a low spatial resolution must be used. Finally, while research has been done to create small and inexpensive screen addons that can turn them into lightfields (such as a pinhole or microlens array placed on top of the screen that constrains the direction of emitted light of any given pixel), there is still an extra hardware component that must be used since unmodified conventional screens have no way to control the direction of emitted light.

There is also research on finding the eye's refractive error with common hardware. NETRA has created an interactive application that puts the human in the measurement loop to evaluate refraction, allowing only a screen and human feedback to be used instead of direct scanning of eye topology with specialized hardware. By using a pinhole array (in a manner similar to the lightfields above) to create points at virtual distances from the eye, the distance between points that should converge at the retina can be approximated using the blur perceived by users. The method works very well for measuring radially symmetric aberrations, but is less applicable for asymmetric or higher order aberrations. While a workaround has been devised for the cylindrical aberrations caused by astigmatisms, complex aberrations like the ones produced by keratoconus might be difficult to capture using their technique.

Methodology Theory

The process of estimating the user's PSF is as follows:

- Initialize $b = \beta b_g + (1-\beta) b_d \in [0,1]^{w \times h}$ where $b_g = \vec{1}$ and b_d is one at its center, 0 elsewhere. Then, initialize A_0 to the 'identity convolution' (a convolution matrix that is one at its center and 0 elsewhere in our case). β is arbitrarily chosen to allow us to represent 'negative radiance' when presenting images to the user and must be higher for people with more severe keratoconus.

- Then, in a loop until the user declares we have reached convergence,
  - Compute $x_i = \arg\min_{x \in [0,1]^{w \times h}} ||A_i * x - b||_2$, the x_i we estimate will be perceived as closest to b by the user.
  - Present x_i to the user, who perceives $A * x_i$
  - The user can toggle between x_i and a solid canvas initialized to $\beta\vec{1}$ and is instructed to draw features from $A * x_i$ onto b_g such that they align, excepting the central dot

(the rationale being that this central dot causes most of the errors, so excluding it allows for the errors induced by it to be compared to the drawn errors). Rapidly toggling between these two canvases lets the user confirm that they drew these features with the correct position and brightness. Note that, since a PSF must sum to 1, we know that $A * \beta\vec{1} = \beta\vec{1}$. Thus, if the user acts perfectly, they will draw $\hat{y}_i = A * x_i - (1-\alpha)(1-\beta)b_d$. However, this is difficult since as the drawing canvas is drawn upon, viewing it will induce more errors as the canvas gets further from uniform gray. Thus we ask the user to only draw the most prominent errors they see in any given step, the image of which we will call y_i.

– For the sake of this process, we assume that y_i is $\hat{y}_i$ in expectation. Therefore, we see $A * x_i = \mathbb{E}[y_i] + (1-\alpha)(1-\beta)b_d$ where we use the sum of noncentral entries in A_i as an estimate for α

$$A_{i+1} = \min_{\hat{A}} \left|\left|\hat{A} * x_i - A * x_i\right|\right|_2 = \min_{\hat{A}} \left|\left|\hat{A} * x_i - (\mathbb{E}[y_i] + (1-\alpha)(1-\beta)b_d)\right|\right|_2^2$$

so we find $A_{i+1} = \arg\min_{A'} ||A' * x_i - (y_i + (1-\alpha)(1-\beta)b_d)||_2^2$ with GD.

Using a monitor that displays based on SRGB means that every image shown to the user must be gamma corrected and every image returned by the user must have its gamma correction inverted to retrieve the relevant radiance.

Next, assume we have properly estimated A in the prior step. Now, for a given image b, we want to find a prior $x \in [0,1]^{w \times h}$ such that $A * x$ is perceptually close to b. Note that finding a true inverse such that $A * x = b$ is likely to be impossible, especially for dark images, due to needing negative radiance. Therefore, we want to maximize $f(A * x, b)$ where f is some function that measures how close images are structurally. For this, we use MSSIM (Mean Structural Similarity Index Measure), which uses the mean of weighted SSIM over patches of the image. The SSIM between two patches $x^{(1)}, x^{(2)} \in [0,1]^{w \times h}$ is made up of three components:

$$Q_{lum} = \frac{2\mu_1\mu_2 + C_1}{\mu_1^2 + \mu_2^2 + C_1}$$
$$Q_{con} = \frac{2\sigma_1\sigma_2 + C_2}{\sigma_1^2 + \sigma_2^2 + C_2}$$
$$Q_{str} = \frac{Cov_{1,2} + C_3}{\sigma_1\sigma_2 + C_3}$$

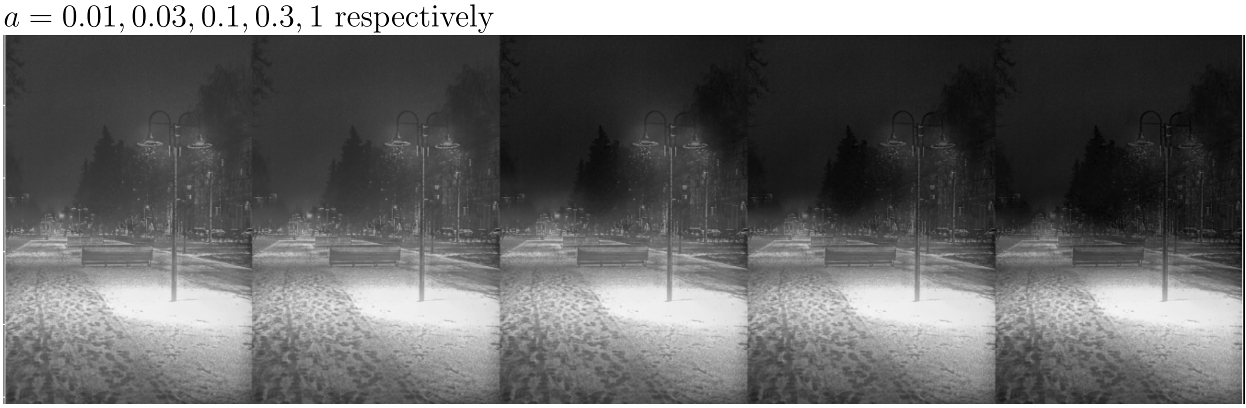
where $\mu_i = \sum_{uv} w_{uv} x_{uv}^{(i)}$ are the weighted means of the images and $\sigma_i = \left(\sum_{uv} w_{uv}(x_{uv}^{(i)} - \mu_i)^2\right)^{\frac{1}{2}}$ are the weighted standard deviations and $Cov_{1,2} = \sum_{uv} w_{uj}(x_{uv}^{(1)} - \mu_1)(x_{uv}^{(2)} - \mu_2)$ where the w's are a gaussian centered on the patch's center (this makes the metric more rotation invariant). $C_{1,2,3}$ are small constants chosen to avoid dividing by 0. Q_{lum} is higher when the brightness of our two patches is close and Q_{con} is higher when they have similar contrast. Q_{str} can be seen as the dot product (or cosine similarity) between the two images after being turned into normalized vectors in $\mathbb{R}^{wh}$. These two vectors being close means that the structure in the two patches is similar, thus is the most important metric for our f since we want to ensure each structure from b is properly represented in our $A * x$. Q_{con} is also important since it ensures our image properly separates adjacent lights and darks. Finally, since we want to sacrifice brightness accuracy for proper structure and contrast, we don't put much importance on Q_{lum}. SSIM is collated from these three metrics by $SSIM(x^{(1)}, x^{(2)}) = Q_{lum}^a Q_{con}^b Q_{str}^c$. We choose $b = 1, c = 2$ and vary a between 0.01 and 0.2 based on whether we want to prioritize keeping darks where we can or whether we want a image that is less inconsistantly 'bloomy' around lightsources (see in analysis). Thus, we return $\arg\max_{x \in [0,1]^{w \times h}} MSSIM(x, b)$ which we find via gradient ascent initializing $x_0 = b$.

Analysis, Evaluation and Results

As someone with keratoconus, using the tool was very frustrating. It's hard to move the mouse with pixel-accuracy as is required, and furthermore, when I got back an image that didn't significantly reduce error (due to slow convergence) I felt very discouraged. Worse, to address a numerical instability issue I was experiencing, I changed the learning rate which adversely affected the convergence speed so badly that I can hardly tell if my inputs were impacting the result.

It was very easy to see errors in my drawing, though. Switching between the two images made it hard to miss even pixel-level disparities. Correcting these errors, however, was a miserable process due to the need to both be at the right place and draw with the right color.

On another note, the image inverter was much more successful. There are a few parameters we can vary in this solver to change our structure/brightness accuracy trade-off. We see that our choice of a for Q_{lum} has lots of impact, especially for small bright lights. The following images have been created by $A * \arg\min_x MSSIM(A * x, b)$ for varying $a = 0.01, 0.03, 0.1, 0.3, 1$ respectively



Observing the image for $a = 1$, we see that it keeps a much lower brightness, but it does so by 'cheating' and erasing the lights that are creating the issue (look at the lampposts, for instance, that have been completely extinguished. On the other hand, for very low a, the image is brightened globally but all light sources are faithfully kept. In the end, I decided to focus heavily on structural detail and use $a = 0.03$ but the actual choice is a matter of preference for the user.

The method ends up significantly brightening images that are dark, but application to images that are already bright tends to not change the image particularly much due to the available radience to subtract from. The following are images where the top is the original and the bottom is the image our method creates where the left is what someone without keratoconus would see and the right is what someone with it (and a kernel matching the kernel we used) would see.

slightly reduced saturation

significantly brightened image

Note that the results we obtain are distinctly better than those from merely raising the brightness of darker features. Above is the image created with a simple linear transform in perceptual color space and below is the result of our method. The ghosting is very obscuring even over the brightened background.

There is a cost to this method, though and that is in the MSE increase caused by so heavily prioritizing structure over brightness. Quantitiatively, this method significantly increases traditional error metrics:

α	$MSSIM_{orig}$	$MSSIM_{after}$	MSE increase
0.6	0.67	0.89	51%
0.45	0.78	0.95	43%
0.3	0.87	0.97	78%

Furthermore, we see that for higher α, corresponding to eyes with more severe keratoconus, the best structure-oriented MSSIM we can achieve drops significantly (all images in this paper use $\alpha = 0.6$). I, however, think this loss is worth the gained sharpness.

Discussion and Conclusion

I definitely would want to retool the PSF guesser in a better way. I think my current approach of using clamped GD to step A is inherently flawed. The main issue I encountered was that, since the point of light is very small, gradient updates using MSE will also be very small. However, the gradients themselves are very unstable due to each weight contributing to every out pixel. In the end, the tool does work as a proof-of-concept interface for a single loop iteration, but I need to seriously rethink how I am propagating A's updates.

One immediately obvious limitation of this method is that we typically look at screens with two eyes which are very unlikely to have the same PSF. Furthermore, this method requires that the user's head be the exact right distance and rotation relative to the screen. This leads to an obvious and applicable usecase for this method - VR. Since there is a screen per-eye and the device is affixed to the user's head, these two drawbacks are immediately addressed and mitigated. However, finding the prior x's that produce b is a slow process, taking dozens of seconds for just a 1000×800 image. The latency constraints on VR pose a large difficulty in this regard.

References

[1] Moschos, Nitoda, Georgoudis, Balidis, Eleftherios, Karageorgiadis, Kozeis, Contact Lenses for Keratoconus- Current Practice, The Open Ophthalmology Journal, 2017

[2] Huang, Wetzstein, Barsky, Raskar, Eyeglasses-Free Displays Siggraph 2014

[3] Pamplona, Mohan, Oliveira, Raskar, NETRA: Interactive Display for Estimating Refractive Errors and Focal Range, ACM Siggraph Papers, 2010

[4] Wang, Bovik, Sheikh, Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Transactions on image processing, 2004

Gallery