ORIGINAL ARTICLE

Detecting topographical eye disorders

Adil Patel*, Rim Gouia-Zarrad†

* Department of Electrical Engineering,
American University of Sharjah, Sharjah, UAE
b00054384@aus.edu
† Department of Mathematics and Statistics,
American University of Sharjah, Sharjah, UAE
rgouia@aus.edu

Abstract—**The goal of this paper is to outline a novel method of detecting topographical eye diseases using a mobile phone. Because the shape of the eye is of concern, its exterior will be evaluated. The description and current implementation of the technique will be discussed, where the keratoconus disease is used as an example. The technique uses pictures taken of the eye using the onboard camera, and each one is compared to idealized images rendered in the phone. The pictures taken will be compared to the idealized images to search for the presence of the disease.**

Keywords-**topographical eye disorder; keratoconus; mobile phone imaging; corneal topography; image processing;**

I. INTRODUCTION

The goal of this paper is to present a novel method of detecting topographical eye disorders which makes use of a mobile phone. The goal is to allow anyone to find diseases in their early stages and warn them to check with a doctor. It works by taking pictures from the camera and comparing them to rendered images of idealized eye shapes. Mobile phones were chosen to provide an accessible method of looking for diseases: they are fast, cheap, and packed with sensors [7]. As

a result, the goal of accessibility has dictated the method developed here.

It was determined that no additional hardware must be connected to the phone in order to make it truly accessible. Many papers such as the one written by Yoon et al [14]; Maeda et al [9]; and Beling da Rosa [5] show eye disease detection methods that require the use of external accessories. On contrast, this method only uses the accelerometers, gyroscope, and camera.

This paper is focused on topographical disorders that affect the shape of the eye, and are not trivial to detect without equipment of some sort. This is because shape variations are on the millimeter scale. This makes them a perfect match for a mobile phone. This method can be applied to the keratoconus disease, which distorts the cornea of the eye into a cone-like structure. Typical shapes are rounded and ellipsoidal in nature. However, the cone-like shape affects refraction of light entering the eye and hampers vision [13]. Figure 1 shows this effect by presenting a map of the sagittal curvature of a diseased eye. The sagittal curvature is defined as a reciprocal of the normal distance from the corneal surface to the optical axis line within the eye. This map is indicative of the shape of the corneal surface. Keratoconus is manifested
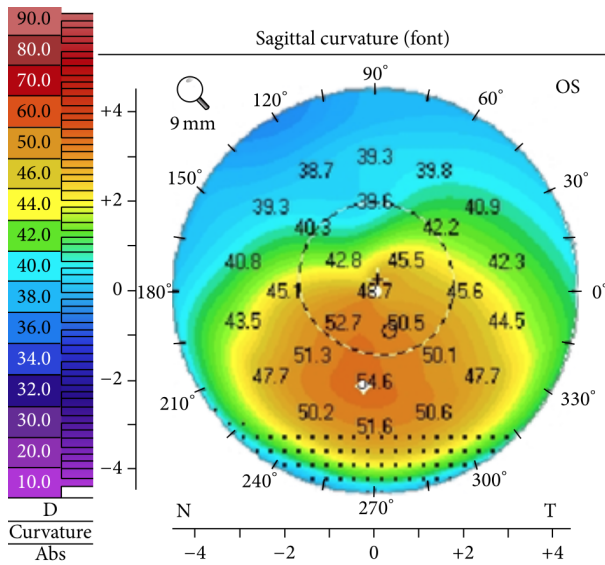
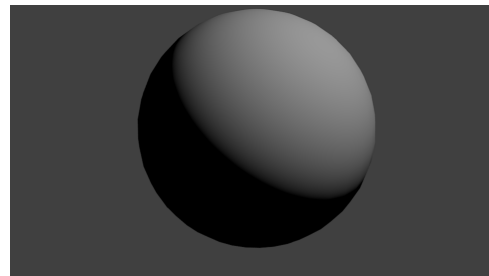Fig. 1. A topographic map of a diseased cornea [12]

Fig. 2. A Sphere with only diffuse components enabled

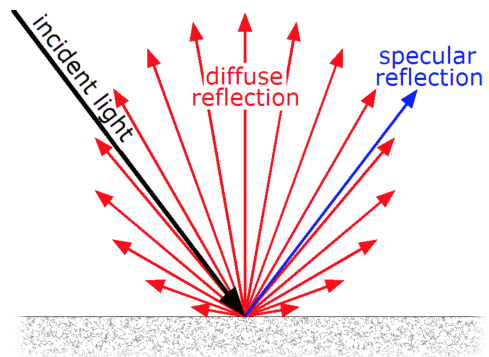
Fig. 3. The scattering patterns of diffuse and specular light [6]

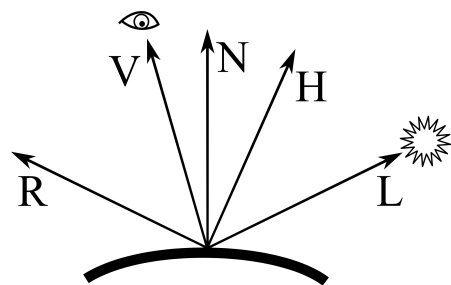Fig. 4. The vectors used in the Blinn-Phong equation [8]

in different ways in these maps; in this example, the bean-shaped red region is an indicator.

Section II will introduce optical terminologies, followed by Section III which has previous work. In Section IV, we will present a novel method for detecting the disorders of the eye.

II. THE BLINN-PHONG LIGHTING MODEL

Due to the fact that this paper involves cameras and computer rendering, it is important to introduce the terminologies used. The computer rendering uses an optical approximation scheme to simulate the interaction with materials and light. This scheme is known as the Blinn-Phong shading model [2]. In this model, light reflected off a surface can be broken into three components: the ambient, diffuse, and specular component. The ambient component is - as the name suggests - the background light of the scene. The diffuse component is caused by light striking the surface and scattering around at multiple angles. This causes a smooth, chalky effect that is visible in Figure 2. Finally, the specular component governs the "shininess" of the surface - the outgoing light scatters in one specific direction. Equation (1) is

the Blinn-Phong shading equation.

$$c = (N \cdot L) \cdot d + (N \cdot H)^{\alpha} \cdot s + a \qquad (1)$$

The values a, d, and s are the colors of the ambient, diffuse, and specular components respectively. In addition, α is the specular component and controls the shininess. The vector N is the surface normal of the material, L is the vector from the light source to a point on the surface, and H is the vector that lies halfway between the

normal and light vector. These vectors can be seen in Figure 4. The output of this equation is a pixel color value. This equation was applied to all pixels being rendered.

III. PREVIOUS WORK

A. *Three-dimensional Reconstruction Methods*

Plenty of literature has been found for three-dimensional shape reconstruction as it is a famous problem in computer vision. To begin, Balzer and Werling describe in [1] the formulation of the problem involving the reconstruction of specular surfaces, as well as some methods. Unfortunately, they require the use of a screen to project fringes, which cannot be achieved with only a mobile phone. Looking further, Crispell, Lanman, et al show how to reconstruct surfaces using only a camera in [4]. However, their method works by detecting depth discontinuities, which do not exist on an eye. Another work by Zhang, Tsai, and Shah that is shown in [15] describes how they did reconstruction with a camera. It used a static camera with a moving light source. This setup does not seem appropriate for the goals, as it would require deliberate and precise movement of light sources. In addition, the approach used by Magda, Kreigman et al shown in [10] can be used for reconstruction, but their setup contradicts the imposed restrictions.

Yoon, Prados, and Sturm show in [14] how they performed three-dimensional reconstruction. Their work is based on multiple photographs of the object at different, known positions. An initial shape was then fed into the algorithm. Their algorithm projected one of the photographs onto the shape, and drew this projection from the perspective of the other images within software. The rendered images and their corresponding photographs are then compared for their dissimilarity. After computing an error value, it was minimized by using variational calculus, and the process was be applied again for different pictures to give an iterative improvement of the shape. This method fits within the requirements of the setup, but is computationally intensive and complex to implement.

B. *Other Methods*

Techniques to scan for eye disorders without any sort of reconstruction have also been used. Maeda et al showed in [9] how they could detected keratoconus using a device called a videokeratoscope. Even though it does require an external piece of hardware, it shows in detail how to classify diseased eye using discriminant analysis. Furthermore, André Luís Beling da Rosa detailed in [5] how to detect and classify diseased eyes using external hardware that projects circular rings onto the eye. The imperfections on the eye shape will distort those projections, which are then detected and analyzed with Zernike polynomials.

IV. CURRENT WORK

As seen above, literature on this problem is scarce. For this reason, we developed a new method that can practically detect the diseases.

A. *Procedure*

The technique presented relies on taking a series of pictures of the eye and comparing them with computer-generated images. The user moves the phone in a circular arc around the eye while taking a series of pictures. At the same time, the accelerometers and gyroscope compute the exact position and orientation of the phone at the instant each photo was taken. Internally, images of an idealized eye are being be rendered from the perspective of the position and orientation data collected at the time of each photograph. Each picture then has a corresponding synthetic image, and they are compared to find diseases. Figure 5 gives a graphical description of this procedure.

Rendering the image requires a high level of accuracy, which is provided by real-time computer graphics techniques. Physically-based shaders simulates materials and light based on measured physical parameters [11]. Coupled with faster hardware, the images render within a few milliseconds [7].

The geometry of the idealized shape does not necessarily have to be a real eye shape. It simply needs to be able to function as an appropriate model. This depends on the type of disease looked
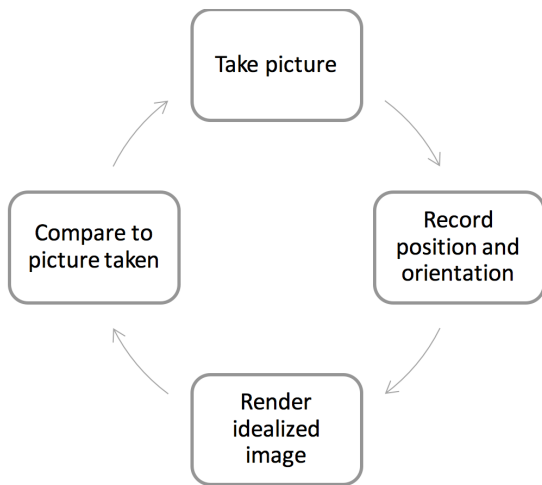
Fig. 5.   The process run when each image is captured

Fig. 6.   The spherical phantom with its triangles highlighted
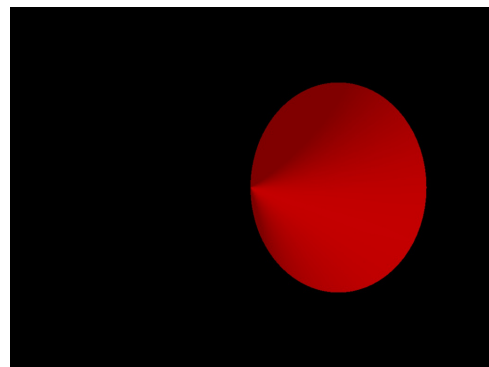
Fig. 7.   The conical phantom used for a diseased eye

for. The implementation shown in the next section elaborates on this point.

B. Current Implementation

The above process has been partially for the sake of simplicity and to verify our models. This has been implemented on an iOS device, using the Metal framework. It can be carried over to other platforms as well. Below is a description of the stages of our implementation.

1) The Shapes Chosen: The first stage was to create phantoms. Cameras typically exhibit non-ideal effects like noise [3]. Phantoms on the other hand, have none of these. They were rendered on the device. For the "idealized shape", a simple colored sphere was used. As described in the introduction, the keratoconus disease is the main focus. This is a disease caused by conical distortions of the eye. As a result, conical shapes were used for the phantoms. Two phantoms were chosen to model non-ideal eye shapes: a cone and a paraboloid. The cone modeled a worst-case-scenario; whereas a paraboloid is qualitatively rounder and less "pointy", and was used to model a moderately diseased eye. All phantoms had a primary color chosen for simplicity. In this case, the color was red.
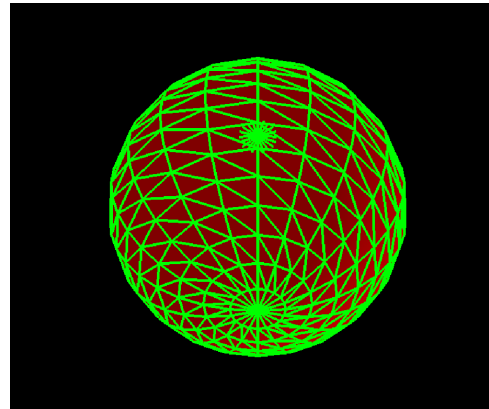
2) Creating the Phantoms: The phantoms needed to be generated on the device before rendering. Since Metal requires triangular primitives to draw surfaces, the phantoms were discretized and then divided into triangular facets. The discretization stage made use of parametric surfaces to divide it up into vertices. This procedure was similar for all phantoms.

After discretizing them, the phantoms were triangulated. Vertices from the discretization stage were given unique indices and grouped together in triplets to form triangles. Three index numbers (that each correspond to a vertex) made up a triangle. Figure 6 shows the final triangulated sphere. The paraboloid and cone were triangulated in a similar manner.

3) Rendering the Images: After creating their geometry, the phantoms were rendered. Rendering

was performed at the native resolution to avoid upscaling artifacts, and anti-aliasing was turned off in order to eliminate fuzziness. These made the phantoms sharp. The shading stage used the Blinn-Phong method as shown earlier to color the pixels. Since iOS devices were used, the sRGB color space was adopted. As such, each pixel was set to an RGBA configuration with 8 bits dedicated to each subpixel component. For the rest of the paper, all colors will be presented in terms of their normalized RGBA intensities. The ambient light had a color of $(1, 1, 1, 1)^T$ with its coefficient set to 0.5. The diffuse color was $(1, 1, 1, 1)^T$ with its coefficient set to $0.3\overline{\cos(\theta)}$ where θ is the angle between the light and normal vector. The overbar denotes the fact that the cosine value was forced to 0 if it happens to be negative. Finally, the specular color was also $(1, 1, 1, 1)^T$ with the coefficient set to 100. Practically, these values will depend on the environment, so there was no universally correct lighting parameters to pick for these phantoms. Thus, the only requirement was to pick sensible and realistic values. In essence, this setup gives the shapes a shiny red material with white lights pointed at it.

To get pixel values for the rendered images, a texture was created and configured as a render target for each frame. After each frame was rendered, a function would lift the data from the center-most pixel in the texture to use it for analysis. The results of this are shown in the next section.

C. Results

Pixel values were read from the render textures in the process mentioned above and used for analysis. Even though four color values were lifted for each pixel, the phantoms only had red color components (see Figure 5). As shown previously, this is because red is a primary color, so only one value per pixel was considered. Hence, only the red values were plotted.

As written earlier, the camera was moved in a circular arc. Therefore, each successive data point corresponds to a larger polar angle. In the simulation, the phantoms were rotated at a constant
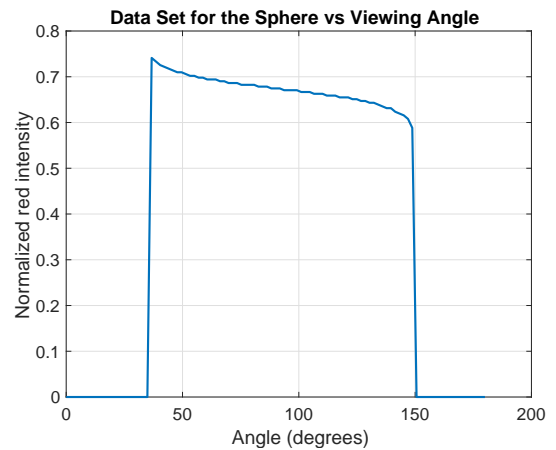

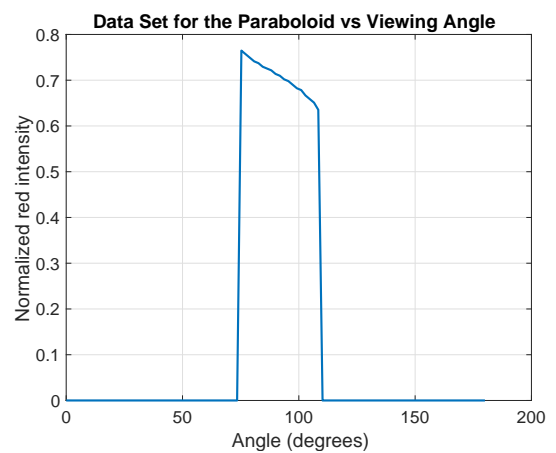Fig. 8. The data for the idealized spherical model

Fig. 9. The data for the mildly diseased paraboloid model

angular velocity; hence, the sample points were regularly spaced.

In all curves, it is first apparent that there are zeros trailing on each side. A zero value means that a black pixel was sampled, so for this to happen, a black pixel must have existed in the middle of the screen. Figure 7 shows that the background was pure black; which means that during the rotation, the background was presented on the middle of the screen. This is because the rotation axis was not aligned with the center of the phantoms. Instead, it was aligned with their apexes to make it more like a real world situation.

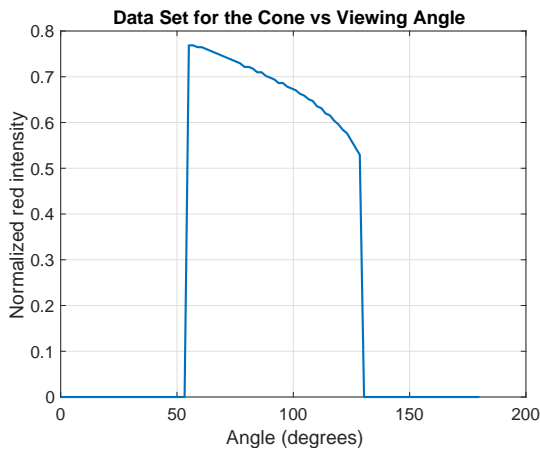In addition, each curve shows a differing inten-

Fig. 10.   The data for the heavily diseased cone model

sity distribution according to the viewing angle. The Blinn-Phong model written above shows that the reflected intensity depends on the material parameters (ambient, diffuse, and reflection coefficients), the light color values, the surface texture colors, and the angle made by the light on the surface normals. Since everything besides the normals was constant, it is apparent that the variations are caused by the geometry of each phantom.

The curves presented have a patterns in the form of their concavities. The sphere has an inflection point in the region of the $90°$ angle. This is very strongly apparent from the graph. The cone, which models a heavily diseased eye, has no inflection point visible in the sampled region of its curve; only in the step discontinuities like all other shapes. Its concavity is strongly negative. The paraboloid, however, has an inflection point in its curve for the sampled region. This point is not as strongly visible as the sphere, which leads to the idea that it lies as an intermediate step between the sphere and the cone. This corresponds to the intended purposes of the curves. It is worth investigating this effect in future works, and applying it to more realistic images of the eye.

These results are relevant to the problem of detecting keratoconus as they present a relation between the geometry of the cornea and the phantoms. The sphere, cone, and paraboloid were in-

tended to model three different degrees of disease. Consequently, it was expected that the sphere and cone exhibit unique characteristics in their intensity curves, with the paraboloid showing something "in-between" the two as the paraboloid was intentionally used to model an intermediate form of the disease.

In summary, the differences in intensity between the different modeled geometry opens the way to a novel method of discriminating between the various keratoconus disease in the cornea.

V. Conclusion

In this paper, we investigated a novel method to detect topographical eye diseases such as keratoconus. It utilizes camera, accelerometers, gyroscope, and rendering hardware of the mobile phone. The implementation was successful in finding a correlation between the varying degrees of diseased eyes. This approach is worth pursuing in future works; using real images and further investigating the correlation between the degrees of disease and the curves fetched.

References

[1] S. Balzer, J.Werling. Principles of shape from specular reflection. *Measurement*, 43(10):1305–1317, 2010.

[2] James F. Blinn. Models of light reflection for computer synthesized pictures. *ACM SIGGRAPH Computer Graphics*, 11(2):192–198, 1977.

[3] I.J. Boie, R.A.Cox. An analysis of camera noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):671–674, 1992.

[4] Daniel Crispell, Douglas Lanman, Peter G. Sibley, Yong Zhao, and Gabriel Taubin. Beyond silhouettes: Surface reconstruction using multi-flash photography. *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 2006.

[5] Andr da Rosa. *An Accessible Approach for Corneal Topography*. 1 edition, 2013.

[6] GianniG46. File:lambert2.gif - wikimedia commons, 2010.

[7] Anthony Gutierrez, Ronald G. Dreslinski, Thomas F. Wenisch, Trevor Mudge, Ali Saidi, Chris Emmons, and Nigel Paver. Full-system analysis and characterization of interactive smartphone applications. *2011 IEEE International Symposium on Workload Characterization (IISWC)*, 2011.

[8] Martin Kraus. File:blinn vectors.svg - wikimedia commons, 2011.

[9] Naoyukl Maeda, Stephen D. Kiyce, Mohammad T. Dehkhoda, and Michael K. Smolek. Automated keratoconus detection with corneal topography analysis. *Cornea*, 12(1):88, 1993.

[10] S. Magda, D.J. Kriegman, T. Zickler, and P.N. Belhumeur. Beyond lambert: reconstructing surfaces with arbitrary brdfs. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*.

[11] Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. Practical physically-based shading in film and game production. *ACM SIGGRAPH 2012 Posters on - SIGGRAPH '12*, 2012.

[12] Yaron S. Rabinowitz. Keratoconus. *Survey of Ophthalmology*, 42(4):297–319, 1998.

[13] Elise A. Slim, Elias F. Jarade, Bilal M. Charanek, Joelle S. Antoun, Adib I. Hemade, Sahar H. Awada, Henry W. Fakhoury, and Carole G. Cherfan. Acute corneal hydrops mimicking infectious keratitis as initial presentation of keratoconus in a 10-year-old child. *Case Reports in Ophthalmological Medicine*, 2015:1–4, 2015.

[14] Kuk-Jin Yoon, Emmanuel Prados, and Peter Sturm. Joint estimation of shape and reflectance using multiple images with known illumination conditions. *International Journal of Computer Vision*, 86(2-3):192–210, 2009.

[15] Ruo Zhang, Ping-Sing Tsai, and Mubarak Shah. Photomotion. *Computer Vision and Image Understanding*, 63(2):221–231, 1996.