RJ 10275 (A0212-006) December 2, 2002
Computer Science

IBM Research Report

Theory for Calibration-Free Eye Gaze Tracking

Arnon Amir, Myron Flickner, David Koons
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Theory for Calibration-Free Eye Gaze Tracking

Arnon Amir
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
+(408)-927-1946
arnon@almaden.ibm.com

Myron Flickner
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
+(408)-927-1776
flick@almaden.ibm.com

David Koons
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
+(408)-927-1952
dkoons@almaden.ibm.com

ABSTRACT

Eye gaze direction and point of regard has proven to be a very powerful and useful source of information in human computer interaction. The user's gaze direction can be used as an input, in addition to the mouse, keyboard and other input means. However, despite the potential applications, gaze tracking systems are still expensive and they require cooperation in calibration. In this paper we develop a theory for eye gaze tracking, introduce the optical plane, and show its analogy to multi-view epipolar geometry, where the eyeball center is considered a focal point. We first use it to prove that eye gaze tracking under fixed head position is a homography between the pupil project center in image plane and the point of regard in screen plane. This theory supports existing gaze tracking methods that require user calibration at each session. Next we use it to develop a family of eye gaze tracking techniques that requires no user calibration, and allow free head motion. The theory is supported by preliminary results of ray tracing simulation, and a prototype system is already in advanced stage.

Keywords

Eye Gaze Tracking, Gaze Detection, Detecting Point of Regard, Pupil, Glint.

1. INTRODUCTION

The idea of using eye gaze tracking for Human Computer Interaction (HCI) is not new. Hutchinson *et. al.* [7] describe a computer system to provide nonverbal, motor-disabled individuals with a means to communication and environmental control. Jacob [8] describes several ways of using eye movements as input to HCI, and Glenstrup [5] also argues that it is possible to use the user's eye gaze to aid the control of a computer application. Recently, Edwards [3] and Lankford [9][10] have proposed development tools that can be used to create eye-aware software applications.

There are many different schemes for detecting both the direction in which a user is looking and the point upon with the user's vision is fixated. Excellent reviews of various eye tracking methods have been published [14][2]. Any particular eye tracking technology should be inexpensive, reliable, unobtrusive and easily learned for it to become widely accepted. The corneal reflection method of gaze tracking is increasing in popularity due to availability of inexpensive cameras and computational power. Commercially available corneal reflection trackers are available from several vendors [1][4][11][13].

Eye gaze tracking technology has proven to be useful in many fields. However, there are three major impediments before gaze tracking is a popular as the mouse as a computer input device.

1. Gaze-tracking technology is still too expensive. This is not a fundamental problem since inexpensive CMOS technology can be applied to the gaze tracking just like it was applied to graphics once well established standard algorithms are deployed.
2. There is no "must have" application. This is a chicken and an egg problem – applications can't be widely deployed and thus become popular until ubiquitous hardware is available. Hardware requires large volumes before it is inexpensive.
3. The usability of current gaze tracking system is inadequate. Users have to run thru a calibration process at the beginning of each session. Current, commercial systems allow very limited head motion without recalibration. Early mice/tablets also needed calibration but eventually the technology became self-calibrating.

This paper concentrates on problem number 3 in the above list. In particular we address the problem of what algorithms can you apply to the gaze-tracking problem to build a system that requires no user level calibration and works with natural head motion.

2. THEORY OF EYE GAZE TRACKING

We start with a brief review of the eye visual and optical model and show the relationship between the eye gaze direction and the eyeball orientation. Next we introduce the *optical plane*, which

gives us insight into the geometrical relationship between the gaze direction and the eye image captured by a camera. We use this model along with known results from multi-view geometry to prove that the mapping between the point of regard and the pupil center is a homography. This supports the classical family of implicit eye gaze tracking methods that are based on fixed head position and direct calibration. Then we introduce a new family of methods which do not require calibration and allow free head motion. We develop three methods, two using stereo cameras, and a second using a single camera, and compare between the various different methods.

2.1 The Eye

We use a classical eye model, as shown in Figure 1. The cornea is modeled as a sphere, its center is on the optical axis. The visual axis, which is the line of sight, is about 5 degrees from the optical axis. This angle may vary among people. It can be measured, stored and used to compensate for the error. In the rest of this section, however, we ignore this angular difference, and assume that the gaze vector coincides with the optical axis. We resort to these assumptions at the discussion and explain their affect on the results, and suggest when and how they can be compensated for.

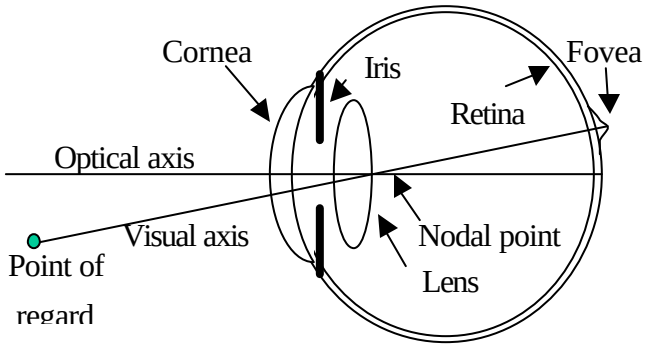

Figure 1: The model of the eye.

2.2 The Optical Plane

Figure 2 shows the configuration of the camera, the eye and the monitor screen in space. We consider a pinhole camera, its focal center is located at point A. To simplify the discussion we assume that the cornea is spherical and that the pupil is a circle on the sphere surface. Let C, r, P denote the center and radius of the cornea ball and the center of the pupil, respectively. We place a point light source on or near to the camera focal point A. We denote it the on-axis light source, to differentiate from any off-axis light sources which might be present for other purposes (see, e.g., [12]). Let G denote the on-axis glint, that is the reflection point of the on-axis light source on the cornea (as it is seen from the camera). Such a light reflection from the cornea is also known as the first Purkinje image. Let g, p denote the projection of G, P into the camera image plan, respectively. We use a perspective projection camera model.

We define the (optical) point of regard on the screen, P', as the intersection point of the line CP (the eye optical axis) with the screen plane. First observe that C, G, g, A are collinear. This is because G is a reflection from a sphere, as it is seen from the direction of the light source. For the reflection of light coming from point A to be seen by the camera at point A, the line AG has to be orthogonal to the tangential plane of the ball surface at point G, which in turn is also orthogonal to the line CG at this point. Hence CG and AG are collinear.

We define the *optical plane* by the three points CGP (connected with dashed lines in Figure 2). This plane passes thru

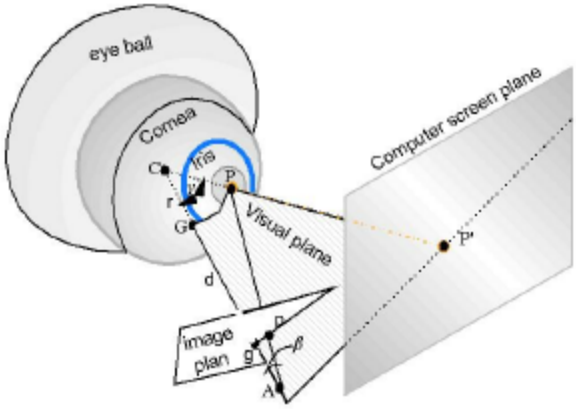

Figure 2: The eye, the camera and the point of regard on the computer monitor (here we ignore the angular

the focal point A and therefore intersects with the camera image plane at line gp. It also intersects with the screen plane at a (dotted) line that passes thru the (optical) point of regard, P'. By definition, the optical plan includes the optical axis of the eye. Hence the points $CGPApg\,P'$ are all coplanar. Note that this is true for any camera, eye and screen position and any gaze direction. This interesting observation is the foundation for deriving the following results.

We now look at the two dimensional configuration of the points on the optical plane, as illustrated in Figure 3. We add the eyeball center, O, which is also on the line going thru CP. The center of the pupil, P, is projected to point p in the camera image plane by a projective transformation. Observe that P is also "projected" to the screen plane by a projective transformation,

where the eyeball center, O, serves as the focal point for this matter. Under a fixed head position the eye center does not move (it is not affected by eye rotation), which implies a fixed projective transformation from pupil center to the point of regard for all gaze directions. We conclude here that point P is being projected to two planes under two a projective transformations.

This particular configuration is well studied in a different context in computer vision. This is the field of multi-view geometry, where points in space are observed by several cameras from different viewpoints [6]. In our case one "view" is not taken by a camera, but is actually the target plane that contains the point of regard, and the eyeball center considered as its "focal point". Thus the optical plane is also an epipolar plane, as it passes thru the two focal points. The line OA, not shown in the figure, is the baseline. Hence the line gp in image is an epipolar line, and so is the corresponding epipolar line defined by the intersection of the optical plane with the screen. We conclude that for a fixed head position, the points P' and p lie on corresponding epipolar lines.

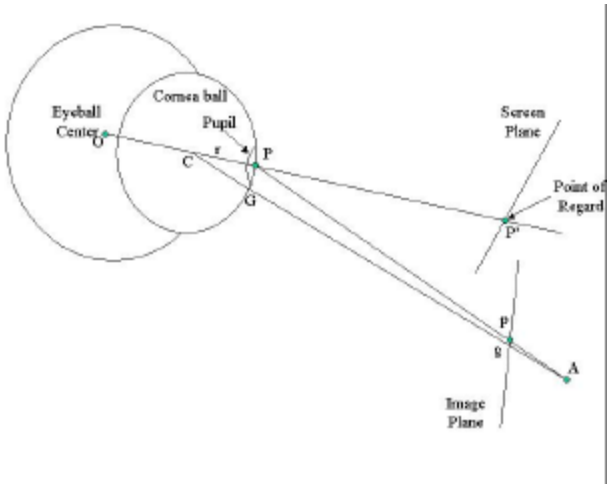

Figure 3 The points on the optical plane. If the head position (and eyeball center) is fixed, then the mapping from p to P' is a homography, where the eye center srerves as the projection point for the screen plane.

2.3 Implicit Eye Gaze Tracking Methods

Most of the existing eye gaze tracking systems use an implicit mapping and direct calibration to detect the point of regard on a computer screen. Note the slight distinction between finding the gaze direction (a vector in space) and finding the point of regard on the monitor (the intersection of this vector with the screen plane). In implicit methods, a mapping is built directly from image measurement of pupil center p (and possibly glint center g) to the point of regard, $P' = f(p)$ on the screen plane. f is a monotone two dimensional function $f : \Re^2 \rightarrow \Re^2$. The three-dimensional gaze vector is not being computed, and the eye location in space is not required. It requires, however, that after the calibration the eye would remain at the same place during the entire session. This implies that no head motion is allowed.

As we showed in Section 2.2, the case of a fixed head position can be formulated using standard epipolar geometry. It follows that the mapping from point p in image plane to point P' in screen plane is a homography. The geometry is given by the fundamental matrix F:

$$p^T F P' = 0$$

where P' and p are given in homogeneous coordinates. The mapping is a homography, and as such it is induced by the selection of a plane in space. In our case this plane can defined by the pupil center at three different gaze directions. The three corresponding points $(p_i, P_i'), i = 1,2,3$ are obtained during calibration process and can be used to directly compute H_p such that

$$P' = H_p\, p$$

However, due to numerical considerations, one would prefer to use more than three points, and to use a robust estimation of the homography. We used nine calibration points, arranged as shown in Figure 3. This is a commonly used calibration pattern for eye gaze tracking systems [13][1] .

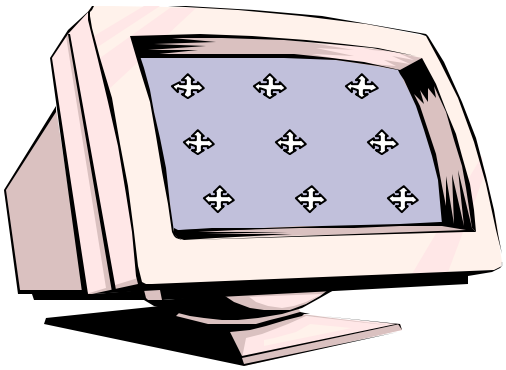

Figure 4: Implicit methods with direct calibration. During user calibration the points are shown to the user one at a time (not required with the new method).

Although the pupil center in image coordinate can be directly mapped to point of regard in screen coordinates, one would prefer to use the glint as a reference point, rather than absolute image coordinates. This is to compensate for small head motion. If the eye was a sphere, then the glint position would be invariant to eye

rotation (or gaze direction), and would be an ideal reference point for small eye translations. Note that eye translation affect the gaze point by the amount of translation, which might be negligible for small translations, while it greatly affect the absolute position of the pupil center in the image (a one inch translation can take the pupil out of the camera field of view). Thus it is crucial to compensate for even slight head motions. The glint is a good reference point for that. Hence we replace $P' = f(p)$ with $P' = f(gp)$, where $gp = p - g$ is the vector from point g to point p in image plane. However the eye is not a spehere, and thus the glint moves slightly with the rotation of the eye. Still, the vector gp is a better choice for computing the point of regard, P'.

In all implicit methods, a calibration process is used to compute the direct mapping parameters. In the calibration process, the user is asked to look at a sequence of targets located at different known locations on the screen. The pairs $(p_i - g_i, P_i')$ are recorded, and the parametric mapping f is derived using LMS or other appropriate minimization criteria. If the fitting does not provide a good estimation of the calibration points, the calibration has to be repeated. This sometimes happened when the user look at a different location then the calibration point, for one or more of the calibration targets.

After the calibration is successfully done, each new measure gp_t is mapped to the point of regard $P_t' = f(gp_t)$. The method is implicit, as it does not determine the eye location in space, nor the gaze vector.

Although this is a very simple method to compute, and is very popular among existing eye gaze tracking systems, it has two main drawbacks. It is sensitive to head position, and it requires user calibration before each session. Note that when the head moves, the mapping f changes. One approach is to compensate for the change by measuring the head motion [13]. Note however that this would require not only the new direction from the camera to the eye, but also the distance to the eye (i.e., relative eye location in space). And it still requires the initial calibration. In order to eliminate the calibration and to allow free head motion we seek different methods.

2.4 The Optical Plane and the Epipolar Constraint

We showed that the optical plane is an epipolar plane. From the image of the eye alone we already have three points on this plane, namely the pupil center, p, the on-axis glint center, g, and the focal point A. These three points provide us with a simple way and a closed-form solution to compute the optical plane, without knowing where the eye is in space. Once the optical plane is found, using the points Agp, it is intersected with the screen plane to produce the corresponding epipolar line on the screen. This step requires a fully calibrated camera, for which the intrinsic (focal length, the principal point, x, y scale factors, and skew) and extrinsic (translation and rotation) are known. Further, we need to know the screen plane position in camera's world coordinates. These however can be measured and calibrated once in a factory, assuming that the camera is attached to the screen and remains at the same position after the calibration.

What remains after we find the epipolar line on the screen is to find where on this line is the point of regard P'. There are several ways to approach this question, and those correspond to different eye tracking methods, all within the family of methods which are based on the computation of the optical plane.

2.5 A Stereo System

In a stereo system, a second camera is positioned at a different location, preferably at the other side of the screen. The two cameras are directed at the user's eye. Each camera produces one optical plane. These two optical planes intersect at a line that is the optical axis, or the (optical) gaze vector. Its intersection with the screen plane is the point of regard P'.

Although the two cameras allow for an explicit computation of the eye location in space (the pupil center is a corresponding point between the two views for which we can compute depth), this is not an essential part of the gaze computation. This simplifies a little bit the computation when both cameras has to track the eye as it eliminates the need to compute the epipolar geometry between the two cameras.

2.6 The Stereo Baseline Method

A variation of the stereo method eliminates the need for highly-accurate positional calibration of the orientation of the cameras by projecting a reference "baseline" onto the cornea. This method uses two illuminators: the on-axis illuminator for one camera serves as the off-axis illuminator for the other camera. The line connecting the two cameras/illuminators serves as a reference line for measuring the relative orientation of the optical plane in the image plane (figure X). Ideally the on-axis glint should be centered on the focal point of the corresponding camera, however the relatively small distortion can be computed. Note that this method does require an estimate of the 3D position of the eye/cornea.

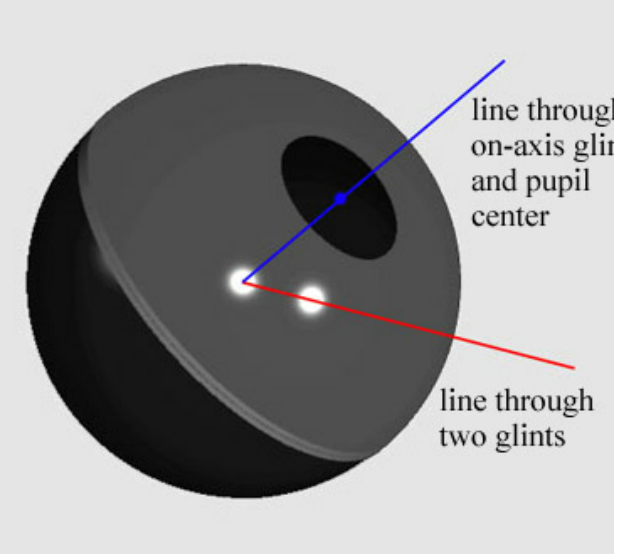

Figure 5: The two glints define the projection of the stereo baseline in image space.

However, both of these stereo systems require two calibrated cameras, and this reflects in increasing system size, complexity and price. Hence we looked at developing a single camera system.

2.7 A Single Camera Solution

From a single camera we get one epipolar line on the screen that passes thru the point of regard. In order to avoid a second camera, one would need to estimate the angle GCP and the cornea center location in space, C. The ray Ag passes thru the cornea sphere center. Therefore, only depth is required to find C. Depth can be measured using different techniques. One way would be by measuring the time-of-flight of the on-axis light from the light source to the eye and back to the camera (as with laser range finders). Depth might also be estimated from finding a few fixed points on the face - points that remain fixed regardless of the facial expression, such as the eye corners and the nostrum. Once the depth is estimated, the vector gp can be back-projected to the eye position. The angle GCP is then computed using the cornea sphere radius, r, which can be measured once per user and stored in the user's profile.

After we find the optical plane, the angle GCP, and the point C, all that is left is to project the optical axis from the eye to the screen, intersect it with the screen, and this is the point of regard.

2.8 A Comparison Between the Methods

The new methods proposed here have two clear advantages over previous gaze tracking methods. They allow completely free head motion – as long as the camera/s are able to track the eye and to detect the features in the image. And, they require no user/session calibration process. The single camera method requires a single time user calibration, to estimate and store the user's eye

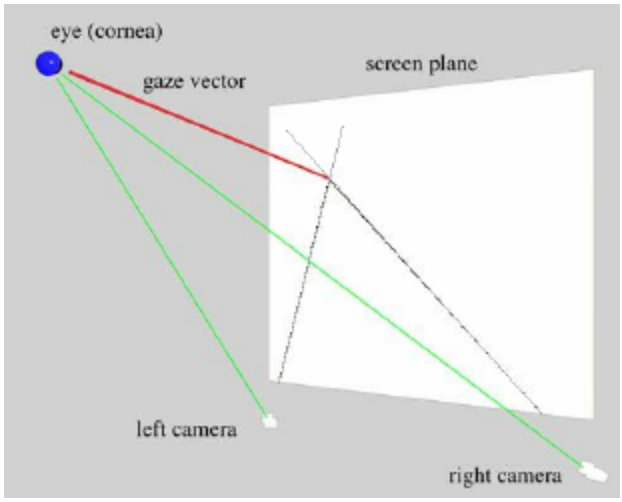

Figure 6: Ray tracing of the stereo configuration. The ga vecetor, the projection lines of the on-axis glints, and t intersection of the two optical planes with the screen a shown.

parameters, such as angular difference between the optical axis and the visual axis, cornea radius etc. But since these do not change between sessions, and since the gaze computation is independent of the head location, the system does not require any session calibration.

However, because these new methods are based on a three-dimensional model of the scene, they do require intrinsic and extrinsic camera calibration, and screen location in world coordinates to compute the intersection of planes between camera image and screen plane. These require a robust camera and monitor setup, and become even more complicate when the camera moves in order to track the eye during head motion.

We now revisit the various assumptions we made along the way and examine their affect on the result. In the above discussion we always compute the optical axis, not the visual axis that actually defines the correct point of regard. This introduces a consistent error in the point of regard. To accurately compensate for this error, one has to compute the eye position in space, as explained for the stereo cases and for the single camera case, and then compute the visual axis at this fixed angular distance from the computed optical axis. Note that with the classical implicit eye gaze tracking methods this error is implicitly taken into account during the calibration process, as the homography that we derive form the calibration points is already compensating for it.

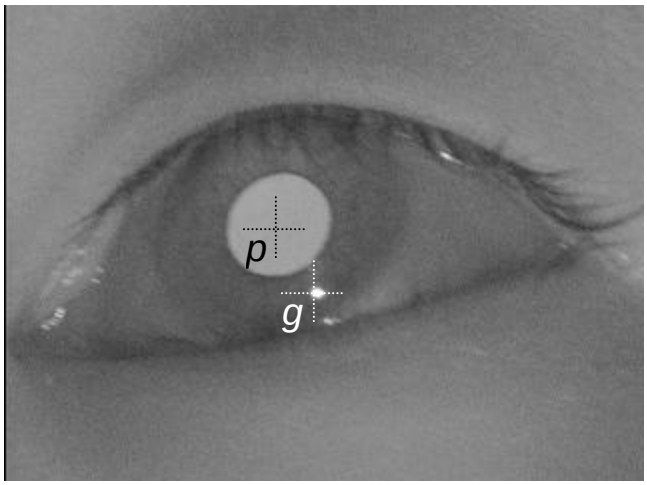

Figure 7: an image of the eye, showing the (bright) pupil and the on-axis glint. The two center points are

We made a few assumptions about the pupil. In particular, we assumed that the cornea is a sphere and we ignored the pupil image refraction when it passes thru the cornea (we later model it in the simulations). Note that the computed optical plane would remain correct even if we relax these two assumptions, as long as the cornea shape has radial symmetry around the optical axis. The effect of refraction would make the pupil center to move along the epipolar line in the image plane. Similarly, the non spherical cornea would cause the on-axis glint not to point to the cornea "ball" center, but rather to some other point along the optical axis. Still, the line that connects the points gp in image plane would be the same line as before, and hence the epipolar line on the screen will also remain correct. However, the single camera method would require more careful computation of the optical axis from the measured points in the image, taking the cornea surface shape into consideration.

Last, although not mentioned before, we assumed that the on-axis glint is reflected from the cornea surface. In extreme eye gaze angles, the glint might fall out of the cornea region and onto the sclera. Again, for the computation of the optical plane this glint is still valid. However, for a single camera method this case has to be treated separately.

3. EXPREIMENTATION

We are currently working on the implementation of these different methods in our lab. The largest practical issue is to get a high-resolution image of the eye AND at the same time to allow for free head motion. This requires a camera with a narrow field of view one hand, and very fast tracking of the eye motion in space on the other hand. We have built the stereo hardware using high speed scanning optics and we are currently working on the camera calibration process.

A typical image of the eye, captured by the eye-tracking camera, is shown in Figure 3. We refer to two points in this image: the pupil center, and the glint center. The pupil is found using the technique described in [12]. The pupil center is then computed as the center of mass of the pupil region. The glint is usually the brightest point in the image, is only a few pixels wide, and is expected to be located inside or near to the pupil region. Its center can be found by computing the center of mass of its region, or by fitting an appropriate model to the gray level pixels.

In order to test of these methods, we used in the meantime some ray tracing simulations. Figure 6 shows the rendering of the stereo configuration. We located the two cameras bellow the screen, which is a natural placement to reduce self-occlusion of the eye by other parts of the face, and is similar to our prototype system. The eye is modeled in a more realistic model than the one we use for the theory above, and includes the refraction of the cornea and its affect on the image of the pupil that is located behind the cornea. The two images of the eye, as obtained by the two simulated cameras are shown in Figure 7.

4. DISCUSSION

5. ACKNOWLEDGMENTS

We thank XXXXXXX for numerous discussions and comments.

6. REFERENCES

[1] Applied Science Laboratories, http://www.a-s-l.com/.

[2] H. Collewijn. Eye movement recording. In R. Carpenter and J. Robson, editors, Vision Research A Practical Guide to Laboratory Methods, chapter 9,pages 245–285. Oxford University Press, 1998.

[3] G. Edwards. A tool for creating eye-aware applications that adapt to changes in user behavior. In Proc. of ASSETS 98, Marina del Rey, CA, April 1998.

[4] Erica Inc., http://www.ericainc.com/.

[5] A. Glenstrup and T. Engell-Nielsen. Eye controlled media: Present and future state. Master's thesis, University of Copenhagen DIKU (Institute of Computer Science), Universitetsparken 1 DK-2100 Denmark, June 1995.

[6] R. Hartley and A. Zisserman. *Multiple View Ggeometry,Cambridge,* University Press, 2000.

[7] T. Hutchinson, K. W. Jr., K. Reichert, and L. Frey. Human-computer interaction using eye-gaze input. IEEE Transactions on Systems, Man, and Cybernetics, 19:1527–1533, Nov/Dec 1989.

[8] R. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. ACM Transactions on Information Systems, 9(3):152–169, April 1991.

[9] C. Lankford. Effective eye-gaze input into windows. In Eye Tracking Research & Applications Symposium 2000, pages 23–27. ACM, November 2000.

[10] C. Lankford. Gazetracker: Software designed to facilitate eye movement analysis. In Eye Tracking Research & Applications Symposium 2000, pages 51–55. ACM, November 2000.

[11] LC Technology Inc., http://www.eyegaze.com/.

[12] C. Morimoto, D. Koons, A. Amir and M. Flickner, "Pupil Detection and Tracking Using Multiple Light Sources", Image and Vision Computing, special issue on Advances in Facial Image Analysis and Recognition Technology, IVC(18), No. 4, March 2000, pp. 331-335.

[13] SensoMotoric Instruments Inc.,http://www.smi.de/.

[14] L. R. Young and D. Sheena. Survey of eye movement recording methods. Behav. Res. Meth. Instrument., 7(5):397–429, 1975.

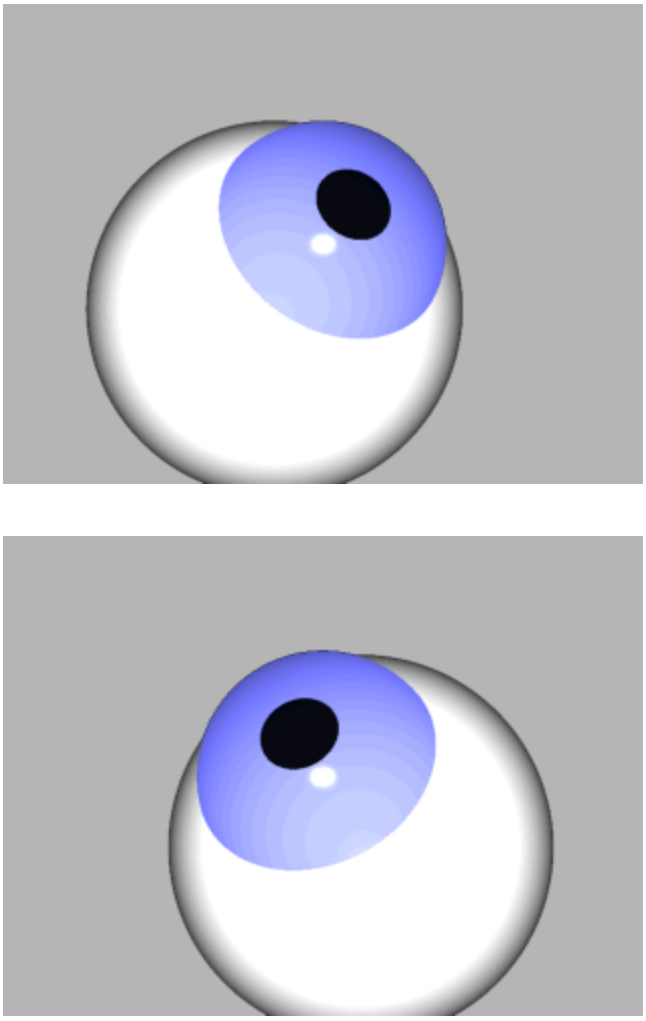

Figure 8: The left and right cameras images of the eye using ray tracing. Glints are produced by corresponding on-axis light sources.