

Gaze Tracking in the Plane

Forest Finnigan



Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-73

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-73.html>

May 17, 2019

Copyright © 2019, 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.

GAZE TRACKING IN THE PLANE

Forest Finnigan

Electrical Engineering and Computer Sciences
UC Berkeley
Berkeley, CA 94720
forestfinnigan@berkeley.edu

May 17, 2019

ABSTRACT

Gaze tracking can be a powerful tool to control a computer or robotic limb. There are many gaze tracking technologies, but vision based gaze trackers, which work off of a standard web cam, have the potential to be quickly accessible. This report assumes that noisy estimates of the head position and gaze direction are known, and then presents a method to find a more stable estimate of the user's gaze on a table plane. The method works by combining estimates of each eye's gaze intersection with the viewing plane. The basic combination achieves fair precision but poor accuracy, so the report describes a simple way to improve accuracy while maintaining good aspects of the system. The system achieves an accuracy of 6.9 cm in the X direction and 5.7 cm in the Y direction for a gaze point experiment.

Keywords Eye tracking · Assistive technology

1 Introduction

This project is meant as a control input to a planar assistive device. The device consists of a large table and a belt system. The device has a place to put the user's wrist. The device can move to a range of set-points, or to arbitrary locations on the table's surface. The device can be controlled by a joystick, or a computer to move the user's hand for a variety of grasping exercises. Ideally, a gaze tracking system would allow another mode of control for the device. Gaze tracking could also offer solo control of the device, as well as more natural control.

The device is currently intended for a clinical environment. Doctors or researchers studying reach to grasp tasks can use the device in combination with other measurement devices. The device is also an intermediate step for a device that will be much smaller, and more portable. This device will offer help outside of the clinical environment and even in general purpose use. The possibility of a small portable assistive device provides extra incentive for a gaze tracking control method.

The gaze tracker will be most helpful for an assistive device if it follows some requirements. The gaze tracker must allow arbitrary placement of the camera and the gaze plane. For instance, the camera may need to be placed on the table pointing at the user, or on a stand closer to the user's face. Gaze trackers that track gaze on a screen do not need to meet this requirement because the camera and screen are always in the same plane.

Another requirement for the gaze tracker is stability. Because the tracker could control a device, it cannot have high variance. Some eye trackers reach better performance in exchange for being very high variance. WebGazer¹, for example, works well when it is being fed calibration points, but quickly becomes very random as time goes on. For this gaze tracker, the opposite may be necessary: accuracy may need to be sacrificed for improved precision.

The gaze tracker must also allow arbitrary head position. There is no expectation that the user will move their head a great amount, but it would be best if the user could move their head some amount without needing to re-calibrate. Arbitrary head position also means that the user does not have to directly face the camera, or put their chin in a rest to use the device.

2 Gaze Tracking Technologies

A gaze tracker detects a user's gaze point or gaze direction to discover where the person is looking. This is in contrast to an eye tracker, which tracks eye landmarks such as the pupil and eye corners. Eye tracking can be useful in its own right as patterns of eye movement can be used for psychological studies. A gaze tracker can use an eye tracker as an

¹<https://webgazer.cs.brown.edu>

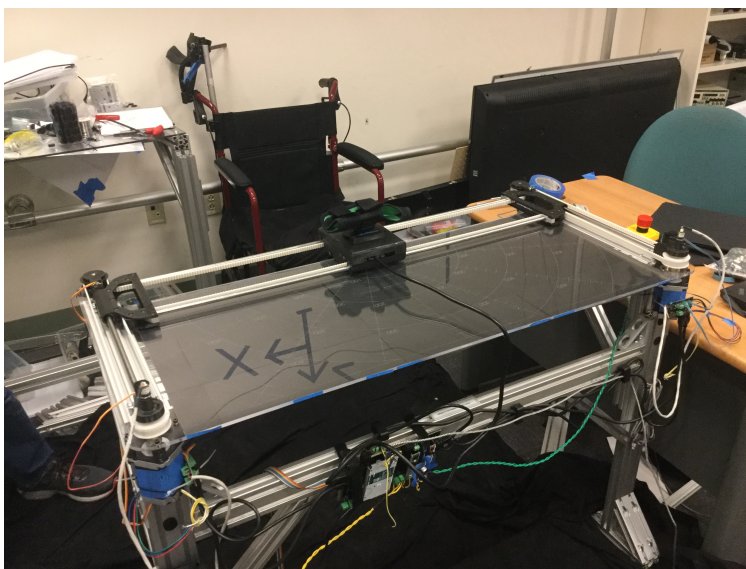


Figure 1: Assistive device in lab.

intermediate step, and perform operations on the eye tracking data to find gaze, but gaze can also be estimated without eye tracking data.

2.1 Infrared Remote Trackers

Remote infrared trackers are currently an established technology. They consist of an infrared camera and can be mounted to the bottom of a computer screen. One example is made by Tobii and is available for consumer gaming. The infrared light produced by the infrared camera reflects most strongly off of the pupil. This makes it very easy to find and track the pupil. Once the pupil is found, a short calibration procedure can be followed to map pupil positions to gaze positions on the screen. For example, the Tobii requires some clicking exercises on the screen.

2.2 Face Mounted Eye Trackers

Face mounted eye trackers consist of a pair of infrared cameras which are mounted to the face. They can also contain accelerometers to assist with tracking. Many virtual reality headsets contain a face mounted tracker to allow foveated rendering. Like the remote eye tracker, the face mounted eye tracker uses an infrared light to detect the pupil. After the pupil is lit with an infrared light, face mounted eye trackers can use filtering and ellipse fitting algorithms to find the exact center of the pupil. This can be directly turned into a gaze direction, and then a gaze point because various measurements like head orientation and camera orientation relative to the eye are already known.

2.3 Vision Based Trackers

Vision based trackers are the most open to research among gaze trackers. There are many open source examples. WebGazer² is an online platform that is calibrated by mouse clicks. Vision based trackers have the special distinction that they rely only on a very ubiquitous technology, color cameras. For the purpose of this project, the user does not have to buy any special equipment to use the system.

3 Past Work

Gaze trackers can be made with or without eye tracking data. Gaze trackers that are made without eye tracking attempt to regress from an image directly to a gaze point or gaze direction. In many cases, the image may be cropped to just show the eye. This is not an unrealistic starting place because there are many very reliable open source pieces of software that do face tracking such as the one created by Tadas Baltrusaitis [1]. Baltrusaitis [1] uses a so called constrained local neural field to find the face landmarks. These would allow any program to easily locate and extract the eyes.

One way to do direct regression is through deep supervised learning. Krafka's work [2] is an example of this approach. The researchers created an app that allowed crowd sourced data collection, and made possible the collection of the large amount of data needed to train a large estimator. Approaches other than deep learning are also possible. Variants of linear regression explored by Lu [3] as well types of non parametric estimation researched by Tan [4] are two examples. The method of non parametric estimation used by Tan is called locally linear embedding and is described in [5]. The hope of these learning based approaches is to learn a low dimensional representation of the high dimensional pixel data. The low dimensional representation might then say something about the gaze.

The other main style of approach starts with eye landmarks. Wang [6] describes a general method for finding the pupil center in 3d space. First the 3d location of the eye and the 2d location of the pupil are found. Again, Baltrusaitis provides a flexible and stable method for finding facial landmarks in 3d space [1]. Many possibilities also exist for finding the 2d pupil location. Timm [7] develops a simple but surprisingly effective method using the color gradient created by the boundary between the iris and the sclera.

Wang's approach continues as follows. In homogeneous coordinates, the projection of the pupil on the image plane (the 2d pupil location) and the 3d pupil location are the same point. This means that there is a line from the camera center, through the pupil projection on the image plane, to the 3d pupil location. Human eyeballs have very little variation in radius. It is also true that the pupil must be on the sphere described by the 3d eyeball center and the average eyeball radius. The last step to find the 3d pupil location is to intersect the eyeball sphere with the line through the pupil. The optical axis is then along the line connecting the pupil center and the eyeball center in 3d. The optical axis hits the gaze plane at the gaze point. Therefore there is a fairly straightforward method to calculate gaze point from 3d eyeball center and 2d pupil position. Figure 2 shows the geometry setup.

²<https://webgazer.cs.brown.edu>

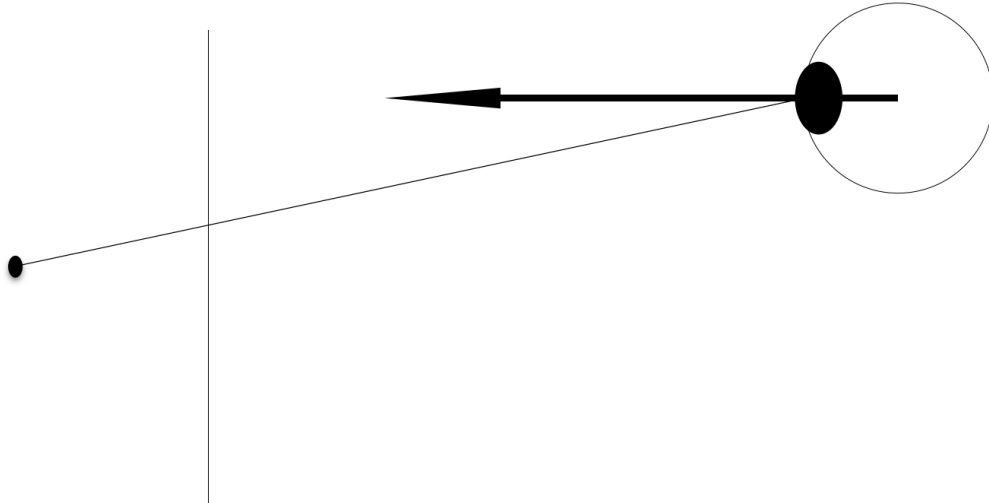


Figure 2: Geometry for pupil.

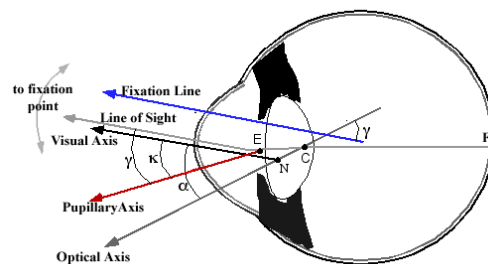


Figure 3: Optical Axis Disparity.

In reality, the optical axis is not the same thing as the visual axis. That means that the optical axis must be shifted by a certain amount to give the true direction that the user is looking. The shift is a user-specific parameter that must be estimated for the most accurate results.

Chen and Ji [8] describe a more unique approach based on weak perspective projection. They use 2d eye landmarks, and do not need any estimates of 3d locations of any landmarks. The use of weak perspective geometry greatly simplifies the calculations from what would need to be used for a perspective camera model. Fortunately, weak perspective will be a close approximation of full perspective projection because points on the face have very similar depth from the camera, especially when compared to the total distance from the camera to the face. As with many of the approaches, Chen uses a calibration procedure. The calibration estimates the discrepancy between optical and visual axis as well as other personal parameters of the user.

Given the usually noisy estimates of eye trackers, it is worth trying to understand the accuracy and precision of gaze trackers. Feit [9] studied the accuracy and precision of two commercial infrared remote trackers. They did tracking in the plane of a computer screen, and found that accuracy and precision were worse in the Y direction, as well as in the lower right of the screen. They also found that performance varied widely for different users, but not for different lighting conditions. They suggested that systems using gaze tracking should aim to measure and compensate for the accuracy and precision of the tracker. For example, if the application senses that the precision has gotten worse, buttons could be made larger, or space efficient hierarchical menus should be used.

Feit [9] also made suggestions for filtering the gaze tracking outputs. They stress the importance of balancing precision with responsiveness. This is because the eyes tend to stay fixed, and then transition quickly to a new fixation point. Trackers that do too much to smooth the signal will miss these quick transitions and slow down the user's interaction with the application. They suggest using a weighted average filter.



Figure 4: Calibration Procedure.

4 Calibration

A calibration step is necessary to locate the gaze plane in 3d space. OpenCV has a method to convert from world coordinates (X,Y,Z) to camera frame which is fixed relative to the camera (x,y,z) . To calibrate, the user places a checkerboard pattern in front of the camera, oriented so that the checkerboard is aligned with the desired world coordinates. The method produces a rotation matrix \mathbf{R} and translation vector \mathbf{t} to map from the world to camera coordinates.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t} \quad (1)$$

The choice of representation for the plane is a point on the plane plus two unit vectors lying in the plane. In the case of this assistive device, the plane is the X Z plane in world coordinates, and two unit vectors in the plane are $(1,0,0)$ and $(0,0,1)$. Also, the calibration procedure places the origin at the bottom corner of the physical checkerboard, which is actually offset from the gaze plane. This means that a point on the plane is actually (a, b, c) where a, b and c represent the shift in millimeters between the origin of the world coordinates, and the desired origin for the gaze plane on the table.

The following procedure will find an equation for the gaze plane in camera coordinates. To move the plane into camera coordinates, the unit vectors, which lie along the plane, must first be rotated. This gives

$$\mathbf{a}_{camera} = \mathbf{R} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{R}_1 \quad (2)$$

and

$$\mathbf{b}_{camera} = \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{R}_3 \quad (3)$$

The point on the plane is converted into camera coordinates as well.

$$\mathbf{c}_{camera} = \mathbf{R} \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \mathbf{t} \quad (4)$$

Finally, the plane is parameterized in camera coordinates.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{a}_{camera}\alpha + \mathbf{b}_{camera}\beta + \mathbf{c}_{camera} \quad (5)$$

5 Creating a Gaze Estimate

5.1 Projecting

For the purpose of this project. The eyeball center and gaze direction will come from an open source eye tracker. The eye tracker will be based on [1] with some modifications to improve the accuracy of pupil tracking. The improvements are based on separate work and are not part of this report.

The gaze tracker will produce estimates of the eyeball center \mathbf{e}_{eye} and gaze direction \mathbf{d}_{eye} . The gaze line is then parameterized as follows.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{d}_{eye}\delta + \mathbf{e}_{eye} \quad (6)$$

To find the intersection of the gaze plane with the gaze line, set (5) and (6) equal.

$$\mathbf{a}_{camera}\alpha + \mathbf{b}_{camera}\beta + \mathbf{c}_{camera} = \mathbf{d}_{eye}\delta + \mathbf{e}_{eye} \quad (7)$$

Rearranging

$$\mathbf{a}_{camera}\alpha + \mathbf{b}_{camera}\beta - \mathbf{d}_{eye}\delta = \mathbf{e}_{eye} - \mathbf{c}_{camera} \quad (8)$$

Putting in matrix notation

$$\begin{bmatrix} \mathbf{a}_{camera} & \mathbf{b}_{camera} & -\mathbf{d}_{eye} \end{bmatrix} * \begin{bmatrix} \alpha \\ \beta \\ \delta \end{bmatrix} = \mathbf{e}_{eye} - \mathbf{c}_{camera} \quad (9)$$

Finally, using the matrix inverse

$$\begin{bmatrix} \alpha \\ \beta \\ \delta \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{camera} & \mathbf{b}_{camera} & -\mathbf{d}_{eye} \end{bmatrix}^{-1} (\mathbf{e}_{eye} - \mathbf{c}_{camera}) \quad (10)$$

In this case, α and β will be used to parameterize the gaze point in the gaze plane, or the plane of the table of the assistive device. Ultimately α and β will describe where in the plane the person is looking, and where they may want to move their hand, while δ is thrown away.

5.2 Combining Gaze Estimates

The intersection of the gaze line with the gaze plane is a noisy estimate of the gaze point. In some situations, the estimate can be very noisy as there may be significant noise coming from the gaze tracker. Also, even if the gaze tracker is not very noisy, even small changes in the angle of the gaze can result in large changes in the projected gaze point. To add to the difficulty, it is also possible for the eye tracker to lose one of the eyes. The eye tracker may lose one of the eyes if the head is turned, part of the face is slightly covered or one side of the face goes into poorer lighting. It is necessary, then, to combine the estimates of gaze point of each eye in a way that greatly reduces noise, while also making it possible to put more emphasis on one eye or the other.

A fortunate fact is that human eyes do not point straight out of the head. In reality, eyes point slightly towards each other, and intersect at the gaze point. In the case of this gaze tracker, the eyes will intersect on the gaze point. This means that the gaze line from the left and right eye should intersect with the gaze plane at the same point.

For the purposed of this gaze tracker, each estimate of gaze point (for left and right eye) is Kalman filtered. The Kalman filter produces not only a mean estimate for the current time step, but also a process covariance. This means that the output of the Kalman filter is a Gaussian distribution representing the likelihood of any gaze point. To find the combined gaze estimate, find the point that maximizes the joint likelihood of the estimates for the left and right eye. The joint likelihood is written as.

$$\mathbf{P}(x_l = a, x_r = b) \quad (11)$$

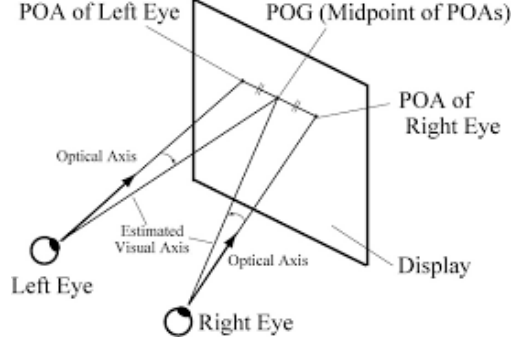


Figure 5: Eyes intersecting on plane.

The left and right eye estimates are independent, so the joint likelihood can be written as.

$$\mathbf{P}(x_l = a)\mathbf{P}(x_r = b) \quad (12)$$

When the probability distributions are Gaussian, the maximum of the joint likelihood function can be written as.

$$\mathbf{x}_{combined} = (\mathbf{P}_l^{-1} + \mathbf{P}_r^{-1})^{-1}(\mathbf{P}_l^{-1}\hat{\mathbf{x}}_l + \mathbf{P}_r^{-1}\hat{\mathbf{x}}_r) \quad (13)$$

\mathbf{P}_l and \mathbf{P}_r are the left and right process covariance matrices as estimated by the Kalman filter. $\hat{\mathbf{x}}_l$ and $\hat{\mathbf{x}}_r$ are the estimates of the left and right eye position as estimated by the Kalman filter. $\mathbf{x}_{combined}$ is the final estimate of the gaze position output by the gaze tracking system. It should be a much more stable estimate of the gaze position.

5.3 Polynomial Remapping

During experimentation, it became clear that performance of this geometric approach was very dependent on the quality of the eye tracker. The open source eye tracker used underestimated gaze angle in many positions. This was apparent in the tracked gaze points. Most points were too close to the center of the screen, with the problem being worse at the top middle and bottom corners of the screen. It seemed that this issue could be fixed somewhat with a polynomial remapping of the tracked gaze points.

To test the idea, I passed the projected points through a polynomial function. The points x and y become x' and y' through

$$x' = a_0x^2 + a_1y^2 + a_2xy + a_3x + a_4y + a_5 \quad (14)$$

$$y' = b_0x^2 + b_1y^2 + b_2xy + b_3x + b_4y + b_5 \quad (15)$$

The user clicks on the screen while looking at the table to create (x,y) and (x', y') pairs. The coefficients are the tunable parameters, which the program finds through linear regression.

6 Contributions

This report makes two specific contributions. The first comes from how the estimates of gaze position for both eyes are calculated. As opposed to taking an average of gaze positions, or using one eye to provide the gaze position, this report uses the uncertainty of each measurement to create a new estimate. By assigning an uncertainty to each eye estimate, and combining the estimates based on that uncertainty, the system is able to favor the most certain estimate in the final estimation of gaze position. This estimation is important because it is possible for one the gaze estimates to be much more noisy than the other, and in these scenarios, the system will benefit from weighing more heavily on the more certain estimate.

The other contribution is in the exact pipeline from eye tracking data to gaze position. Many other approaches choose a geometric pipeline or a regression style pipeline. Geometric style pipelines aim to do some form of exact geometrical

calculation. Geometric approaches may have tunable parameters, which are estimated through calibration, and have specific meanings in the geometric setup (distance between eyes, eyeball diameter and so on). Regression based approaches aim to make an approximate calculation. Regression approaches also have tunable parameters, but those parameters do not have specific meaning to the problem. This report makes an exact geometric calculation followed by a regression step. The geometric calculation finds the general orientation of the gaze point, while the regression estimates an exact calibrated position on the gaze plane.

7 Experimental Setup

The goal of the project is to allow arbitrary visual tracking on the plane of the table. The technique is as follows. The camera is placed at the end of the table facing towards the user. Then, calibration is performed as described in the report. The user sits behind the table as if they were using the assistive device to perform a reaching task. The user then tracks a rectangle around the edge of the reaching space with their eyes. A dot on the computer screen indicates where on the rectangle the user should look as the experiment progresses. Because the user looks at the table and not the computer screen, a second experimenter indicates where the user should look at every time step. More qualitative results are gathered in a second gaze point experiment. In this experiment, the user looks at 10 radially spaced points on the table, as indicated by the program, similar to the rectangle experiment. In both experiments, the program records and compares desired and actually measured gaze points.

The user needs to train the system before each of the remapping experiments. The user trains the system by looking at a point on the gaze plane and clicking on the corresponding point in the program's interface. For these experiments, the user enters 25 training points spaced evenly on the gaze plane. The training points are not points on the gaze rectangle, or the radially spaced points from the gaze point experiment. This means that there is some difference between the training and testing points.

There are a couple of considerations to make about the use of the eye tracker. The eyes do not naturally track lines across a table as is needed in this experiment. The eyes naturally look at points or follow moving objects. For the purpose of this experiment, the user can use their finger or a pointer to track along the rectangle, and then follow the moving object with their eyes.

Like many gaze trackers, this gaze tracker is affected by head motion. While part of the aim of this project was to create a gaze tracker that worked with arbitrary head motion, it is very difficult to completely remove all effects. To try to understand these effects, the user will operate the tracker in three modes. Under natural tracking, the user will try to pretend they are not being tracked and will simply trace the rectangle with their eyes. With stationary or static tracking, the user will hold their head in a comfortable position and try to move as little as possible while tracking the points. For exaggerated tracking, the user will exaggerate natural head movements and try to point their face at the tracking point.

A few more miscellaneous assumptions of the experimental process are as follows. Lighting during the experiment is chosen to be good. The user's face is well lit from the front. Also, lights behind the user may lead the camera to darken the user's face, so there are no bright lights behind the user. The user is an experienced user of the system. For calibration, because there is a lot to juggle, the user will also have an experienced assistant.

8 Results

The results for the three experiments for a single user are as follows. The tracked rectangle is in yellow, and is 30 cm tall and 45 cm wide. The entire black surface represents the extent of the plane of the table, and is 50cm tall and 90 cm wide. The white circles represent the points used in the gaze point experiment. The colored dots are the gaze points produced by the system. Ideally the colored points would all sit on the rectangle.

For the gaze point experiment, the mean error and standard deviation of the each gaze point was measured. The average of those values for polynomial remapped, and not polynomial remapped are as follows in Table 4. The values of the remapping polynomial were also recorded and displayed in Table 3

Table 1: Outputs

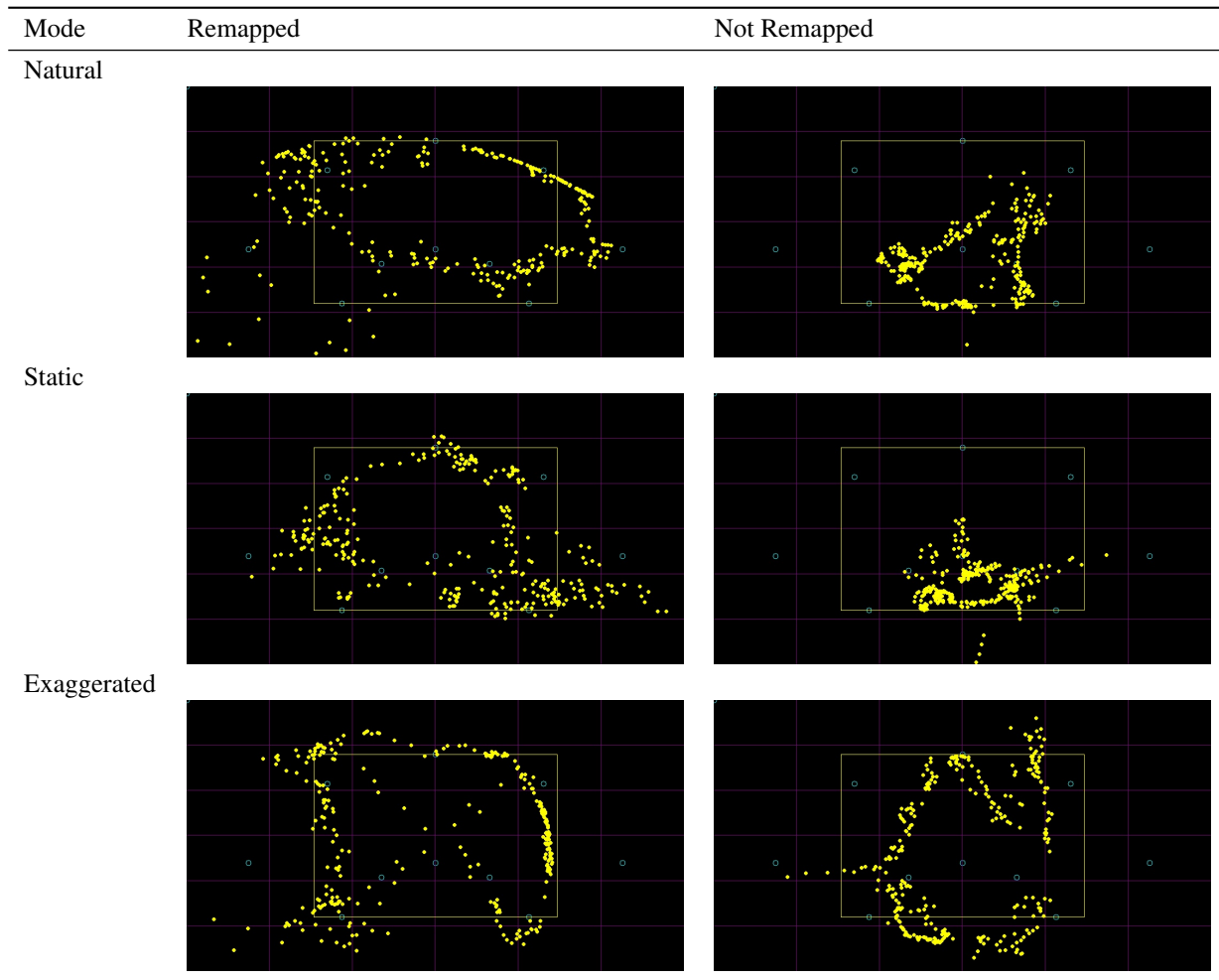


Table 2: Performance For Rectangle

	Remapped (cm)	Not Remapped (cm)
X err natural	7.5	7.4
Y err natural	11	9.4
X err static	7.6	10
Y err static	6.9	12
X err exaggerated	6.1	7.9
Y err exaggerated	6.1	8.2

Table 3: Polynomial Values

Mode	a_0	a_1	a_2	a_3	a_4	a_5
Natural	-0.0028	-0.0011	.0057	.13	-3.2	85
Static	-.011	-.0015	.013	4.16	-4.3	-280
Exaggerated	-.0008	-.0067	.0015	.15	-2.1	82
	b_0	b_1	b_2	b_3	b_4	b_5
Natural	-0.0028	-.0044	.0055	10	-60	-530
Static	-.015	-.0029	.0081	7.4	-1.2	-480
Exaggerated	-.0079	-.00087	.00080	4.2	.16	-150

Table 4: Performance For Points

	Remapped (cm)	Not Remapped (cm)
X err	6.9	24.1
X std dev	7.3	3.0
Y err	5.7	9.7
Y std dev	4.1	1.9

9 Discussion

The system with the remapping does to do a better job of following the desired rectangle. This is true for natural, static and exaggerated head movements. There is also the added benefit that if performance is lower than desired, the system can be tuned further. Of course, the downside of tunability is that the system needs to be tuned before use. Also, the quality of the tracking will be based on the quality of the tuning.

For the gaze point experiment, the remapping greatly increases the accuracy. There is a decrease in precision, but this trade-off is more than worthwhile. To provide a concrete comparison, the remapped results for the gaze point experiment gave accuracies and precisions of several centimeters. Feit [9] found accuracies and precisions of below one centimeter. In those experiments the researchers did tracking on a computer screen with of the shelf remote infrared gaze trackers, but also did tracking with a variety of lightings and users. In contrast this report did tracking in a very large 90 cm plane with a vision based tracker, but also had good lighting and an experienced user.

The coefficients for the polynomials are similar in orders of magnitude. However, the coefficients do vary considerably from experiment to experiment, which is not ideal. Another limitation is noise. While the system can be quite stable for much of the experiment (in the case of the rectangle), there are moments also when the system becomes very noisy.

A major next step will be to address the issues with the polynomial remapping. It is important to understand when and why the coefficients vary. Also, it is important to develop a training procedure that is quick and consistent. Improvements could involve keeping some pre-computed values, or performing regularization on the coefficients. Another next step is to reduce noise in the system. A first approach is to filter the gaze point more aggressively, possibly with a linear filter. A larger step to reduce noise and stabilize the gaze point is to include logic to decide when the user has fixed their gaze, and only move the gaze point when a new point of fixation is found.

References

- [1] Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Robinson. Constrained local neural fields for robust facial landmark detection in the wild. In *International Conference on Computer Vision Workshops (ICCV), 2013*. IEEE, 2013.
- [2] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, and Wojciech Matusik Antonio Torralba Harin Kannan, Suchendra Bhandarkar. Eye tracking for everyone. In *Conference on Computer Vision and Pattern Recognition (CVPR), 2016*. IEEE, 2016.
- [3] Feng Lu, Yusuke Sugano, Takahiro Okabe, and Yoichi Sato. Inferring human gaze from appearance via adaptive linear regression. In *IEEE International Conference on Computer Vision, 2011*. IEEE, 2011.
- [4] Kar-Ha Tan, David Kriegman, and Narendra Ahuja. Appearance-based eye gaze estimation. In *IEEE Workshop on Applications of Computer Vision (WACV), 2002*. IEEE, 2002.
- [5] Sam Roweis and Lawrence Saul. A probabilistic approach to online eye gaze tracking without explicit personal calibration. In *Science, 22 December 2000*, volume 290, pages 2323,2325. IEEE, 2000.
- [6] Kang Wang and Qiang Ji. Real time eye gaze tracking with 3d deformable eye-face model. In *International Conference on Computer Vision (ICCV), 2017*. IEEE, 2017.
- [7] Fabian Timm and Barth Erhardt. Accurate eye centre localisation by means of gradients. In *Sixth International Conference on Computer Vision Theory and Applications, 2011, Proceedings of*, 2011.
- [8] Jixu Chen and Qiang Ji. 3d gaze estimation with a single camera without ir illumination. In *International Conference on Pattern Recognition, 2008 19th*. IEEE, 2008.
- [9] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringe Morris. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *CHI Conference on Human Factors in Computing Systems, 2017 Proceedings of*, pages 1118,1130. ACM, 2017.