Abstract—This project deals with tracking eye gaze location and predicting future eye movement sequences using classical machine learning and Hidden Markov Models, respectively, for users wearing head mounted displays (HMD). Eye gaze location can enhance the virtual reality experience inside an HMD in two ways: foveated rendering and retinal blur. Both of these techniques require accurate eye gaze location predictions to prepare the next frame in a video. The eye gaze location prediction can be accomplished by first tracking and estimating the present eye gaze location and then using the past locations to predict the future locations. This is a joint project done to satisfy CS229’s and CS221’s course project. The work shown in this report explains different approaches using Hidden Markov Models to predict future eye gaze locations once the present location has been estimated via our tracking approach. The approach to track the eye gaze location constitutes the CS229 portion of this project and is summarized briefly at the beginning to connect the two projects. For a much more in-depth analysis of the tracking approach, we refer the reader to the CS229 analog of this report.

I. INTRODUCTION

In this project we perform eye gaze tracking using standard machine learning techniques (regularized linear regression, SVR) and then predict the future eye gaze location using techniques like Hidden Markov Models. The ability to predict eye gaze accurately can have a huge impact in Virtual Reality (VR) Headsets since it gives the developer flexibility to enable foveated rendering or retinal blur to enhance content and have an immersive VR experience. Eye gaze prediction also answers the question of what content to adjust when the user is blinking. Both retinal blur and foveated rendering require high accuracy (so that the scene is rendered and blurred at the right locations) and low latency (so that the content is updated perceptually in real-time). Current state of the art is capable of high accuracy, but latency is still a huge bottleneck, especially in foveated rendering. Our approach to eye gaze tracking and prediction differs from traditional signal processing intensive methods, such as the ones that use Kalman filters.

II. PROBLEM SETUP

It has been studied that eye motion can be modeled into three primary states: fixation, pursuit and saccades as described in [1]. In fixation (F), a person is looking at a stationary object for a prolonged time interval and hence the eye gaze velocity is almost negligible. In pursuit (P), eyes are tracking an object slowly in a scene with some velocity in a deterministic manner. In saccade (S), motion of the eye is erratic and often in a straight line, but the velocities are extremely large and there is no content consumption i.e. brain doesn’t process any information during a saccade.

For the tracking stage, we rely on the image of the eye and do not need to know the eye movement. But when we want to predict the future eye gaze location, which is the scope of this project it becomes essential to model the eye motion. The approach followed in [1] uses an HMM to predict the future eye gaze state with the hidden variable being F, P or S. It then uses linear prediction to estimate the eye gaze location. The approach taken in this project is different as we use the inherent eye motion model but want to predict the eye gaze itself using the Hidden Markov Models without using the linear predictor with the expectation being that the approach would be more generic. We use two different approaches to model our hidden variable and compare the results with a kalman filter implementation and finite difference models.

III. TRACKING APPROACH

The eye gaze can be characterized by an $x$ and a $y$ location. We use various techniques ranging from linear regression to support vector regression to neural networks to track the eye gaze and they are outlined briefly here as they are part of the Machine Learning Project. But the output of this pipeline is the eye gaze locations and this forms an input to the AI project.

We needed to generate a dataset for the tracking approach because we could not find a dataset that contained labeled images of the eye from a side-view. However, this dataset was not used to evaluate the prediction approach because the images of the eye were taken seconds apart and so no motion of the eye was captured. Thus, we used a popular dataset for attention prediction [2] to evaluate our prediction models.

While there is an oracle for the dataset of the tracking approach, there is not one for the dataset of the prediction approach. This is because it is not possible to collect ground truth eye motion because any collection method uses an eye tracker (or sensor) that will have inherent noise. Hence, we have treated the data from the [2] paper as the oracle when we evaluate our error metrics. This assumption is made because we believed their eye tracker, with an error of 0.5° of the viewing angle, was sufficiently accurate.

A. LEAST SQUARES WITH REGULARIZATION

Least squares was used as a baseline for eye gaze tracking. In least squares, the $x$ direction is estimated independently of the $y$ location using the following model $p = Xw$ where $p$ is the eye gaze location, $X$ is the design matrix and $w$ is the weight vector. For regularized linear regression, the objective function is the L2 norm:

$$J(w) = ||p - Xw||_2^2 + \mu ||w||_2^2$$
The best way to select the model parameter $\mu$ is to plot the optimal trade-off curve between the two objectives which allows us to visualize the effect of changing $\mu$. We use the regularized version as we want to avoid overfitting the training data and hence by adding a constraint on the weight vector we are able to achieve our objective. As we independently calculate errors for $x$ and $y$, they are converted into the mean squared error and finally into viewing angle error in terms of the viewing direction.

**B. SUPPORT VECTOR REGRESSION**

The $x$ and $y$ locations can also be estimated via support vector regression. The problem, in the primal domain can be written as:

$$
\begin{align*}
\min_{w,b} & \frac{1}{2}\|w\|^2 \\
\text{subject to} & \quad y^{(i)} - w^T x^{(i)} - b \leq \epsilon \quad \text{for } i = 1, \ldots, m \\
& \quad w^T x^{(i)} - b - y^{(i)} \leq \epsilon \quad \text{for } i = 1, \ldots, m
\end{align*}
$$

To use this model, one must select a tolerance, $\epsilon$, and a hyperparameter for the slack variables, $C$, and then the weights are learned to minimize the objective. This problem can be solved more efficiently in the dual domain, and is explained more thoroughly in [6]. Like SVM, support vector regression can also be kernelized.

**C. NEURAL NETWORKS**

Another machine learning model we used for eye-tracking was an Artificial Neural Network (ANN). The software library used to design, train, visualize, and simulate the network was MATLAB's neural network framework. The network formed attempts to solve an input-output fitting problem with a two-layer feed forward neural network. This consists of a hidden layer with a sigmoid transfer functions and an output layer with a linear transfer function. The algorithm used to train the network is Bayesian Regularization [3]. The effect of the number of neurons in the hidden layer was studied by training and testing the network using a different number of neurons in the hidden layer. A diagram of the final neural network selected is shown in Fig. 1

![Fig. 1. Neural Network](#)

**IV. CURRENT APPROACHES**

The current approaches for predicting eye gaze location are developed on top of principles such as maximum likelihood estimation, Kalman Filters, and Hidden Markov Models. Our work builds off of the Hidden Markov Model, but we include some other work for comparison.

**A. FINITE DIFFERENCE MODELS**

While no paper uses a finite difference model exclusively, many papers like [1] use models that are extensions of the finite difference model; hence, we have decided to cover it as it serves as our baseline.

A finite difference model estimates the eye gaze's trajectory based on approximations of the derivatives calculated from the observed eye locations. A *first order finite difference model (FOFDM)* estimates the trajectory of the eye based solely on the first derivative: velocity. The calculation of the velocity is a backward difference

$$
\tilde{v}(x_t) = \frac{x_t - x_{t-1}}{\Delta t}
$$

where $\tilde{v}(x_t)$ is the velocity at position $x_t$ and $\Delta t$ is the time interval between $x_t$ and $x_{t-1}$. The prediction of the next location with the FOFDM can be calculated as follows

$$
\tilde{x}_{t+1} = x_t + \tilde{v}(x_t)\Delta t
$$

The *second order finite difference model (SOFDM)* is similar to the first order model except it uses the first and second derivative to estimate the trajectory. The backward difference for velocity is the same as above. The backward difference for acceleration is

$$
\tilde{a}(x_t) = \frac{x_t - 2x_{t-1} + x_{t-2}}{\Delta t^2}
$$

where $\tilde{a}(x_t)$ is the acceleration at position $x_t$. The prediction of the next location with the SOFDM can be calculated as follows

$$
\tilde{x}_{t+1} = x_t + \tilde{v}(x_t)\Delta t + \frac{1}{2} \tilde{a}(x_t)\Delta t^2
$$

**B. KALMAN FILTER**

Usually Kalman Filters are used in conjunction with other methods, e.g., using Kalman filters with HMMs [7]. Rather than covering the comprehensive list of all possible applications of the Kalman filter, we simply cover its main two uses.

The Kalman filter has two forms, the filtering form and the predictive form. Both are used all across the literature. The filtering form is usually used to de-noise the current estimate, while the predictive form is used to predict the next state based on the past statistics and past observations. We have implemented the predictive form of the Kalman filter for comparison against the baseline and our approach. The method is summarized in [4].

**C. FPS HMM**

The fixation, pursuit, and saccade Hidden Markov Model (FSP HMM), is one current approach [1] that models the state of the eye as either in the fixation state (not moving), pursuit (slow moving) or saccade (fast random moving). The idea of this approach is to have a linear dynamical system for each state, $S$, that will be enabled if the Hidden Markov Model predicts that $S$ will be the next state. One drawback to this approach is that it assumes the saccades are random and hence very difficult to predict. Thus, the state space
is reduced to just fixation and pursuit. This model is very simple and intuitive, but lacks the complexity to handle the complex motion of the eye.

We draw a lot of inspiration from this approach and we attempt to build on this approach by considering other states that perhaps more closely model the motion of the eye, e.g., position and the angle of the velocity.

V. PREDICTION APPROACH

Our models of prediction depend on the framework of Hidden Markov Models. Much like [1], our approach is also to model the true motion of the eye as a hidden random variable. Where our models differ is that our observed variables are (noisy) measurements from the eye tracker and the hidden variables are the true (noiseless) measurements, as opposed to [1] which uses the hidden variables as indicators for signaling which linear dynamical system the eye movement is currently governed by\(^1\). An illustration of a discrete HMM as a Bayesian network is shown in Fig. 2.

![HMM Illustration](image)

Fig. 2. An HMM with hidden variables \(H_i\) which each emit one discrete observed variable \(E_i\). Here the variable \(t\) is an integer that indexes time.

As Fig. 2 suggests, each variable \(H_i\) and \(E_i\) are distributed as follows:

\[
H_i \sim P(H_i|H_{i-1}) \\
E_i \sim P(E_i|H_i)
\]

When a Bayesian network has the form in Fig. 2 and has variables distributed as above, it is called an HMM. For the rest of this report we will refer to \(P(H_i|H_{i-1})\) as the transition probability and \(P(E_i|H_i)\) as the emission probability.

An HMM is completely specified by giving the hidden and observed variables as well as the transition and emission probabilities. First we will cover the positional HMM which uses eye gaze location as variables and then we will cover the angular HMM which uses the angle of the eye gaze’s velocity as variables.

A. POSITIONAL HIDDEN MARKOV MODEL

The first Hidden Markov Model implemented uses true eye gaze locations, i.e., \((x, y)\) positions, in the scene as the hidden states and the sensor output as the noisy observations, i.e., \((\tilde{x}, \tilde{y})\) positions, of these hidden states. We assume that an \((x, y)\) pair can be modeled independently, which means there is one HMM for the \(x\) state and another for the \(y\) state. This allows the emissions to be modeled simply as gaussian distributions about the true location. The dataset that we used captures eye gaze locations at 30 frames per second and hence is enough to model the fixation and pursuit modes as described in [1] since a continuous smooth motion can be observed. It cannot however be used to model the saccade motion as that requires a frame rate of more than 100 fps. For each of the two HMMs i.e. for \(x\) and \(y\) we also need the transition and emission probabilities to completely characterize the model. These are discussed in detail hereafter.

1) Hidden and Observation State Space: The hidden and observed state space for the positional HMM is specified by the pixel locations in the scene and the domain for each \(x\) and \(y\) models is governed by the size of scene in the respective direction. An important point to be noted is that we only consider integer states and round off our observations to have a discretized space which can be operated upon i.e.

\[
x, \tilde{x} \in [1, N_x] \quad \text{and} \quad y, \tilde{y} \in [1, N_y]
\]

where \(x, y, \tilde{x}, \tilde{y} \in \mathbb{Z}, N_x\) is number of columns and \(N_y\) is number of rows.

2) Transition Probabilities: It is easy to estimate the transition probability using the given sequence of eye gaze locations. This has to be estimated independently for each video which makes it content specific but also improves the prediction for that content \(^4\). The following equation is used to learn the transition probabilities:

\[
p(x_i|x_{i-1}) \approx \frac{\# \text{ of times } h_i = x_i \text{ and } h_{i-1} = x_{i-1}}{\# \text{ of times in state } h_{i-1} = x_{i-1}} \quad (4)
\]

The same equation with variables \(y_i\) and \(y_{i-1}\) is valid for the other \(y\) HMM.

3) Emission Probabilities: We assume a gaussian distribution to model the sensor noise with some mean and standard deviation as given in (5) and use this as the emission probability for each of the states. The prior we assume on the data is as follows:

\[
p(e_i = \tilde{x}_i|h_i = x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu)^2\right) \quad (5)
\]

where \(\sigma^2\) is the variance.

4) Prediction: Prediction can be accomplished by simply conditioning on the evidence

\[
P(H_{t+1} = h_{t+1}|E_1 = e_1, \ldots, E_t = e_t) \propto \sum_{h_t \in D(H_t)} P(H_t = h_t|E_1 = e_1, \ldots, E_t = e_t) p(h_{t+1}|h_t) \quad (6)
\]

B. ANGULAR HIDDEN MARKOV MODEL

The second Hidden Markov Model used for prediction uses true discretized directions (or angles) of the eye gaze’s velocity as hidden variables, \(H_i\), and the estimation of the eye gaze’s velocity from the eye tracker as the evidence

\(^1\)This is also called a switching linear dynamical system.

\(^3\)Also called "evidence" in some literature.

\(^4\)One of the possible use cases would be prediction for highly popular videos with views in hundreds of thousands on youtube and be able to use the output of this algorithm for content specific compression.
variables. The major assumption behind using the direction of velocity is that the underlying motion of the eye is smooth, i.e., continuous and differentiable. This of course depends on the refresh rate of the eye tracker. If the eye tracker refresh rate is too slow, then each subsequent estimation of the eye gaze location and direction will appear random. Thus, this model requires that there is some temporal correlation between eye gaze location measurements.

Like the positional HMM, there are four things we must include to fully characterize the model: the hidden state space, the observation or evidence state space, the transition probabilities and the emission probabilities.

1) Hidden and Observation State Space: The hidden and observed state space for the angular HMM, or the domain of \( H \) and \( E \) (respectively), is specified by \( N_\theta \in \mathbb{N} \), which is the number of equally spaced directions that the eye gaze’s velocity is binned into. The fixation state, specified by \( S_f \), is included into the state space and it represents all velocities whose magnitude does not exceed \( s_f \in \mathbb{R} \), where \( s_f \) is in pixels/sec. Thus, any state space for an angular HMM is specified by a tuple \((N_\theta, s_f)\) and can be written as

\[
D(H_i) = D(E_i) = \{\theta_1, \theta_2, \ldots, \theta_{N_\theta - 1}, \theta_{N_\theta}, S_f\}
\]

where \( D \) is the domain of a random variable.

![Fig. 3. Example state spaces for parameterized by \((N_\theta, s_f) = (4, 1)\) (left), \((N_\theta, s_f) = (8, 1)\) (middle) and \((N_\theta, s_f) = (16, 1)\) (right)](image)

Fig. 3 has an illustration of possible state spaces parameterized by \((N_\theta, s_f)\). As an example, if \((N_\theta, s_f) = (4, 1)\), then \(D(H_i) = D(E_i) = \{\theta_1, \theta_2, \theta_3, S_f\}\) where

\[
H_i = \begin{cases}
\theta_1 & \text{if } -45^\circ \leq \angle v_i \leq 45^\circ \text{ and } |v_i| > 1 \\
\theta_2 & \text{if } 45^\circ \leq \angle v_i \leq 135^\circ \text{ and } |v_i| > 1 \\
\theta_3 & \text{if } 135^\circ \leq \angle v_i \leq 225^\circ \text{ and } |v_i| > 1 \\
\theta_4 & \text{if } 225^\circ \leq \angle v_i \leq 315^\circ \text{ and } |v_i| > 1 \\
S_f & \text{if } |v_i| \leq 1
\end{cases}
\]

A similar expression holds for \( E_i \) except all \( v_i \)’s are replaced with \( \tilde{v}_i = \tilde{p}_i - \tilde{p}_{i-1} \) and \( \tilde{p}_i \) are the measured eye gaze locations from the eye tracker.

2) Transition Probabilities: Just like with the positional HMM, we learn the transition probabilities. When learning the transition probabilities, we assume that emission variables are deterministic, i.e., \( P(E_i | H_i) = 1 \) if \( E_i = H_i \). Thus, we can estimate the transition probabilities \( P(H_i | H_{i-1}) \) as the frequency of landing in state \( h_i \) from \( h_{i-1} \), where \( H_i \) is the random variable and \( h_i \) is the value \( H_i \) takes on. As an example,

\[
p(h_i | S_f) \approx \frac{\# \text{ of times } h_i = \theta_1 \text{ and } h_{i-1} = S_f}{\# \text{ of times in state } h_{i-1} = S_f}
\]

3) Emission Probabilities: Since the ground truth for the measured eye gaze velocities is unknown, we must enforce a prior for the emission probabilities of the eye gaze location. While a Gaussian prior is standard, it works well when the domain of the random variable has a linear ordering, which we do not have in this case. While there are adjacent states, e.g., \( \theta_j \) has neighbors \( \theta_{j-1} \) and \( \theta_{j+1} \), there is also the fixation state, which is also adjacent to all \( \theta_i \). There are many choices to assign the fixation state emission probability, but we have chosen to assign it a fixed probability of 0.5, i.e., \( p(e_i = S_f | h_i = \theta_j) = 0.5 \). As for the emission probabilities of the other \( \theta_j \) states, we assume that they are simply Gaussian distributed. Since the \( \theta_j \) states \( \theta_1 \) and \( \theta_{N_\theta} \) are adjacent, the Gaussian distribution needs to take this into account as well.

We can express this Gaussian as follows

\[
p(e_i = \theta_m | h_i = \theta_n) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(f_{N_\theta}(m,n))^2\right)
\]

where \( \sigma^2 \) is the variance and

\[
f_{N_\theta}(m,n) = \begin{cases}
|m-n| & \text{if } |m-n| < \frac{N_\theta}{2} \\
\frac{N_\theta}{2} & \text{otherwise}
\end{cases}
\]

Lastly, the emission probability, with \( S_f \) as the hidden state, is assumed to be uniform, i.e.,

\[
p(e_i = \theta_m | S_f) = \frac{1}{N_\theta + 1}
\]

4) Prediction: Prediction can be accomplished simply by conditioning on the evidence as given in (6)

VI. RESULTS

To test our approaches, we used the dataset in [2]. While we could have used the dataset we generated for evaluating the tracking approach, we thought it would be more convenient to use a popular dataset so that our results could be compared. The dataset consists of 15 subjects watching 12 videos two times (or trials). To generate the eye tracking data, the subject’s wore a head mounted eye tracker that operated at 30 FPS. The mean error in the eye tracker was around 0.5° of the visual angle with a standard deviation up to 0.45°. For this dataset, a 2° viewing angle corresponds to 32 pixels (with reference to the image) 5. Because the ground truth eye gaze locations are unavailable to us, we can only compare our results to the measured eye gaze location from the eye tracker used in [2]. Thus, it is possible to achieve an error smaller than the error specified by the eye tracker 6, but this simply means our prediction algorithm is predicting the next measurement very well; it says nothing about how much we are outperforming the eye tracker.

5 The paper [2] says 64 pixels, but the meaning of "pixels" there refers to the Samsung color monitor used to collect the dataset, which is double the number of pixels of the actual video.

6 Mean of 0.5° with standard deviation up to 0.45°
Since we suspected that the learned transition probabilities would be content specific, we generated a HMM model for each of the 12 videos. The results of the positional HMM and angular HMM are not exactly comparable since the positional HMM error is in pixels, but the angular HMM error is a false classification rate. We attempt to make comparisons when appropriate.

For both the positional and angular HMM, each video’s HMM model is trained on 12 subject’s data (both trials) and tested on 3 subject’s data (both trials).

A. RESULTS: POSITIONAL HMM

Using the values as mentioned in the dataset we use $\mu = 0$ and $\sigma = 15$ to define the emission probability and predict the next eye gaze location. Figure 7 illustrates how the emission probability looks like for each of the x and y HMMs. The values obtained in terms of x and y, i.e., euclidian distance, are used to calculate mean squared error and then converted in terms of viewing angle error in degrees and are tabulated in Table [I]. The baseline for our project was the first order model which uses a first order finite difference model that approximates the motion of the eye with only the velocity estimated by the current and previous time-step.

For the Kalman filter implementation, we used $Q = 16^2$, which is proportional to $1^\circ$ in viewing angle. We also used, $R_0 = (0.45 * 16)^2$ for the variance of the eye tracker.

As can be seen from the table, the positional HMM model performs better than Kalman filter implementation (as described in [4]) and the SOFDM prediction. The HMM model also outperforms the FOFSM prediction in 10 out of 12 videos. This is because the salient content in the last two videos is roughly static in reference to the video frame. This means there were not many saccades and the average pursuit velocity was much lower than the other videos. The reason this results in a good prediction via the FOFSM, is that the small, slow motion is smooth and smooth trajectories have derivatives that can roughly be approximated via a first order approximation.

The SOFDM prediction performs worse on all videos primarily because the frame rate is not high enough. Because eye gaze location begins to decorrelate very quickly in time, the eye gaze location can appear random if the frame rate was too slow. The difference between the SOFDM and the FOFSM is that acceleration in the SOFDM uses two points in the past to calculate the trajectory, whereas the FOFSM only calculates the velocity, which uses one point in the past. Saccades are brief, and are the primary cause of error in the prediction because they appear random when the frame rate is not high enough. Because saccades are brief, their error propagates in a finite difference model according to the order of the model since that is the number of future predictions that will depend on a “random” measurement.

Also, since the 2nd order approximation assumes there is an acceleration term, which means once a subject saccades, a 2nd order approximation will project a trajectory where the eye has sped up even more despite the fact that fixation usually follows saccade [5]; not a faster saccade.

Analysis was carried out for different values of standard deviation as well (for the positional HMM) and the output is illustrated in Figure 8. This matches our intuition that when the sensor output reading becomes noisier the prediction gets worse, i.e., the error in prediction increases. Figure 4 illustrates eye gaze prediction as compared to the observed gaze locations.

B. RESULTS: ANGULAR HMM

For the following results we set $\sigma = 5$, which is the average velocity of fixation we observed in the first dataset (See Fig. 6). Using the variance of fixation is an appropriate choice for the variance of the emission since any error in
the fixation measurement is proportional to the error in the eye tracker and the intrinsic movement of the eye. Fig. 6 is a plot of one subject's eye gaze measurements watching the bus movie that illustrates the average velocity of the three states: fixation, pursuit and saccade.

We also set $s_f = 1$px/sec, which corresponds to a low estimate of typical velocities in the fixation state. This parameter is intrinsically tied to the emission probability. When $s_f \gg 0$ then the emission probability $p(e_i = S_f|h_i = \theta_j)$ is likely not 0.5. This parameter does not effect the validity of our results, merely the success rate for classification.

The first result we will show is the success rate of classifying the velocity angle for each future time-step; this is compiled for the first few parameterizations of the state space in Table II. A successful classification occurs when

$$\angle e_{t+1} = \arg \max_{h_{t+1}} P(H_{t+1} = h_{t+1}|E_1 = e_1, ..., E_t = e_t)$$

The success rate is essentially the proportion of frames in either the training or testing set that were successfully classified into one of the states in the angle state space. A more compact representation of Table II is the average success rate over all videos vs. the parameterization of the state space, which is shown in Fig. 5. The figure shows that with the parameterization when $s_f = 1$ results in a roughly linear increase in error as $N_\theta$ increases.

![Fig. 5](image)

VII. NEXT STEPS

There are a couple of avenues we did not have time to explore fully, but we felt were worth mentioning.

1) An interesting extension to either the Positional or Angular HMM is to clump two states into one, i.e., multiplex the states. For example, the measured states could be combined to make a new state such that $E'_t = (E_t, E_{t-1})$. While the transition probabilities for these multiplexed states can be learned in the same fashion as for the non-multiplexed states, the emission probabilities are now very different because they depend on multiple variables. It is then possible to learn the emission probabilities by assuming that the observations follow a mixture of gaussians. While it is clear how to do this for the positional HMM, applying this to the angular HMM is not straightforward.

2) While training and applying the HMM model to the data, we discovered that it was possible to estimate not only the next state, $H_{t+1}$, from the current evidence, $E_1, ..., E_t$, but also the next few states with reasonable accuracy. This could be done by repeatedly applying the transition probabilities to the current posterior probability, and taking the argmax to determine the most probable next state. We observed that the next state could only be reasonably estimated for about 5 to 8 time-steps (at 30 fps) before reaching a state that did not change.

VIII. CONCLUSIONS

The purpose of this paper was to design different prediction models and investigate other state of the art approaches. Our analysis shows that it is possible to achieve error on the order of the eye tracker error, simply with a positional HMM. It also shows that depending on the application and the complexity desired, the FOFSM can perform comparably to the position HMM. Still, in terms of accuracy, the positional HMM performs best amongst the models we implemented.

---

TABLE I

ERROR IN VIEWING ANGLE OF POSITIONAL BASED PREDICTION MODELS

<table>
<thead>
<tr>
<th>Model</th>
<th>bus</th>
<th>city</th>
<th>crew</th>
<th>flower</th>
<th>foreman</th>
<th>hall</th>
<th>harbour</th>
<th>mobile</th>
<th>mother</th>
<th>soccer</th>
<th>stefan</th>
<th>tempe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pos. HMM</td>
<td>Train</td>
<td>0.727</td>
<td>0.749</td>
<td>0.749</td>
<td>0.750</td>
<td>0.734</td>
<td>0.708</td>
<td>0.727</td>
<td>0.831</td>
<td>0.757</td>
<td>0.750</td>
<td>0.876</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.707</td>
<td>0.755</td>
<td>0.747</td>
<td>0.745</td>
<td>0.673</td>
<td>0.695</td>
<td>0.759</td>
<td>0.856</td>
<td>0.685</td>
<td>0.741</td>
<td>0.694</td>
</tr>
<tr>
<td>FOFDM</td>
<td>N/A</td>
<td>0.928</td>
<td>0.897</td>
<td>1.355</td>
<td>0.934</td>
<td>0.879</td>
<td>0.781</td>
<td>0.876</td>
<td>0.883</td>
<td>0.851</td>
<td>0.807</td>
<td>0.691</td>
</tr>
<tr>
<td>Kalman</td>
<td>N/A</td>
<td>3.50</td>
<td>3.25</td>
<td>4.78</td>
<td>3.51</td>
<td>3.16</td>
<td>2.81</td>
<td>3.27</td>
<td>3.19</td>
<td>3.09</td>
<td>3.11</td>
<td>2.55</td>
</tr>
</tbody>
</table>

---

This is due to the fact that the limit of the transition matrix raised to a power converges.
In fact, it should also be possible to improve our results even further by cascading a Kalman filter to the output of an HMM to de-noise it.

Another takeaway is that the angular HMM performs reasonably well for up to 32 angle states. While its results are not directly comparable to a viewing angle, if implemented properly, it could still greatly improve the bandwidth allocation for a video.

ACKNOWLEDGMENT

We want to thank Professor Percy Liang and our assigned TA Justin Fu for their valuable feedback on the proposal and progress report and for being available for discussions and giving insights into approaching the problem in a systematic manner.

REFERENCES


APPENDIX

![Example Emission Probability for the Positional HMM with σ = 15](image)

![Plot of velocity of the eye gaze location from neighboring frames from the first subject viewing the bus movie](image)
Fig. 8. MSE varying sigma for Positional HMM

Fig. 9. Mixture Of Gaussian for emission probability estimation

Fig. 10. Predict w/o emission