Tertiary Eye Movement Classification by a Hybrid Algorithm
I-VDT-HMM

ABSTRACT

The proper classification of major eye movements - saccades, fixations, and smooth pursuits - remains essential to utilizing eye-tracking data. There has been difficulty separating out smooth pursuits from the other behavior types, in particular smooth pursuits from fixations. We propose a new algorithm, I-VDT-HMM, for tertiary classification of eye movements. The proposed algorithm combines the simplicity of two foundational algorithms, I-VT and I-DT, as has been implemented in I-VDT, with the statistical predictive power of Hidden Markov Models, in particular the Viterbi algorithm. We evaluate the fitness across a dataset of eight eye movement records at eight sampling rates gathered from previous research, with a comparison to the current state-of-the-art using the proposed quantitative and qualitative behavioral scores. The proposed algorithm achieves better results than the comparison algorithms in clean, high sampling frequency eye tracking data at the cost of classification time.

CCS CONCEPTS

• **Human-centered computing** → *Interaction techniques*; • **Theory of computation** → *Pattern matching*;

KEYWORDS

I-VDT-HMM, Tertiary Eye Movement Classification, Hybrid Algorithm, Fixations, Saccades, Smooth Pursuits, I-VDT, HMM, Viterbi, Probability, Threshold

1 INTRODUCTION

In order to further advance eye tracking research and push forward the use of eye trackers in industry, the identification of eye movements is highly desired. To date there are six primary types of eye movements exhibited by the human oculo-motor system (HOS): fixations, saccades, smooth pursuits (SPs), optokinetic reflex, vestibulo-ocular reflex, as well as vergence [8]. Of these, fixations, saccades, and SPs are the most frequently studied. Intuitively, fixations are what one experiences when staring at a stationary object. Due to such, it can be easily classified using an overall relative velocity of zero degrees per second, or alternatively using a net zero

movement over a period of time. As noise is likely to be present in most eye movement samples, algorithms that employ single threshold classification, especially if the assumption is made that fixations will have zero velocity, may require adjustment on a sample by sample basis. Saccades are where one quickly moves their eyes from one object to another, commonly exhibiting speeds in excess of 300 degrees per second [1]. Thus, classifying saccades based off of a high overall velocity between points can be an effective method. SPs are a function of the HOS that can be described as when ones eyes attempt to maintain high acuity on a moving object [2, 9]. Occasionally one would get distracted or lose focus of the object, in which case a catch up saccade may be exhibited to regain focus. As SPs have variability in their speed, identifying them while also in the presence of fixations can be difficult. This is especially pertinent when noise is present in the data. As fixations are exhibited when someone is focused on an object, they are frequently used in human-computer interaction applications as a selection method [5]. Additionally researchers have found abnormalities in saccade and SPs that have led to the diagnosis of some pathologies of the HOS [4]. Thus the continued development of methods for the classification of these eye movement types is still thought to be an important area of research.

I-VT and I-DT are foundational threshold algorithms used to separate fixations from saccades. I-VT uses a velocity threshold that takes advantage of the large distinction in velocity between the fast moving saccade and the relatively stationary fixation [12]. I-DT uses two thresholds to make the distinction: duration and dispersion. The duration threshold can be dependent on the users and the stimulus. However, the threshold is commonly used with a minimum limit due to the HOS's inability to pick up information in less than 100 ms [14]. Dispersion is the sensitivity of the algorithm in regards to an eye movement's position. I-DT places a window the size of the duration threshold over a series of eye movement points. In the case of a saccade, the points would break the dispersion threshold over that window and be classified as such. Under ideal conditions, the dispersion threshold for a fixation would be zero. However, as noise is inherent in eye movement classification, assuming the threshold is zero can lead to poor results. Classifying saccades from fixations is effective when using approaches that search for single features such as I-VT and I-DT. However, classifying the two from smooth pursuits is not possible using these methods alone. Two algorithms, I-VVT and I-VDT [7], build off of the previous foundational algorithms and incorporate an additional threshold to make this classification. Both I-VVT and I-VDT use an initial velocity threshold to classify between saccades and both fixations and SPs. The remaining points are then classified either using another velocity threshold in I-VVT or using a dispersion threshold as in I-VDT.

Threshold algorithms have the ability to provide promising results. However, the thresholds used in the algorithms tend to vary

based on the input data. In some cases eye trackers provide substantially more noise than others, in which case dispersion and velocity thresholds should be updated accordingly. Thresholds aren't the only method in eye movement classification, however. Other methods previously used in the field include classification using the main-sequence relationship, amplitude-duration relationship, saccades' wave form [8], or statistical probabilities [13].

Hidden Markov Models (HMM) are statistical models that attempt to discover hidden states. The forward algorithm, also known as the Viterbi algorithm [3], is used to determine the most likely sequence of hidden or unobserved states. This is done by calculating three probabilities: the observational probability of each of n states, the transitional probabilities between each state, and the probability at each time step that any of the previous states will lead to the current state. At the beginning of the algorithm two matrices are created: a traceback matrix and an emission matrix. The emission matrix is populated at each step by determining the maximum probability given some transition. The traceback matrix is populated by determining which of the n state probabilities is higher at a given time step. Following the completion of the algorithm, an iterator will travel from the last column index in the traceback matrix, starting in the row with the highest probability, and go backward through the traceback matrix using the value of that cell. Effectively, the Viterbi algorithm works by choosing the series of classifications which has the highest probability [3]. An approach to automatically classify binary eye movements using a HMM has been noted [12] and has been appropriately named I-HMM.

Behavioral scores provide an automated method of creating meaningful classifications when using a step-ramp stimulus [6]. The scores are created under the assumption that the saccade and fixational stimulus are encoded to follow a normal HOS's behavior. It follows that the selected thresholds will hold the same performance when given different stimulus. As the equipment is unlikely to change following the calibration procedure, behavioral scores allow the automated selection of thresholds for classification immediately after calibration.

As eye tracking is a quickly growing field, meaningful improvements in the classification of eye movements continues to be a sought after goal. In this paper we propose a new hybrid algorithm, I-VDT-HMM, which builds off of the previous work of I-VDT [7] and I-HMM [12] in an attempt to take the advantages of a threshold algorithm while statistically enhancing our results using the Viterbi algorithm [3]. To determine the benefits of our algorithm in a variety of conditions, we subsampled our high quality eye tracking data into 8 subsample frequencies and tested it across 8 subjects. Our results are compared against the state of the art I-BDT algorithm [13], as well as the algorithm ours is based off of, I-VDT [7]. Our *MATLAB* implementation of I-VDT-HMM can be found at https://github.com/BerndtSam/I-VDT-HMM.

2 EVALUATED EYE MOVEMENT CLASSIFICATION ALGORITHMS

2.1 I-VDT

I-VDT is a seminal algorithm in tertiary eye movement classification [7]. The algorithm is a combination of the I-VT and I-DT algorithms,

using a velocity threshold to identify saccades, while a moving dispersion window is used to separate fixations and smooth pursuits. I-VDT is noted for fast evaluation time, accurate classification, as well as ease of implementation [7].

2.2 I-BDT

Bayesian Decision Theory Identification, I-BDT, is a probability based algorithm designed for low resolution eye trackers proposed in [13]. I-BDT requires no calibration as it is based entirely on eye positional data and so is operational at run time. I-BDT uses a bayesian decision theory approach where it relies on a prior and likelihoods to calculate the posterior probability of a classification using velocity and a movement ratio over a temporal window as classification features. I-BDT uses the assumption that the velocity and movement ratio of a fixation must be zero, which will be discussed later. A more robust description of the algorithm can be found in [13].

2.3 I-VDT-HMM

I-VDT-HMM is the hybrid algorithm proposed in this work. It is derived from I-VDT [7] in that it uses a velocity threshold to separate out saccades, and a dispersion window to separate fixations from smooth pursuits. In order to statistically ensure the resulting scores, two two-state Viterbi algorithms are employed; once after I-VT and the other after I-DT. The first HMM, like I-VT, is used to separate saccades from a combination of both fixations and smooth pursuits. In this HMM iteration, the velocity of each state is used as the feature classifier. This works well as both fixations and SPs have relatively the same velocity when compared in a noisy environment, and saccades tend to be much faster than the two. The second HMM is ran after the dispersion window separates fixations from smooth pursuits. This HMM once again has two states; one for fixations and the other for smooth pursuits. In this HMM iteration, similar to I-DT, the dispersion of each class is employed as the feature classifier. Thus rather than using the velocity to make classifications, for each point you create a dispersion window and until that window is broken, you sum up the total dispersion. Upon completion of each HMM, we determine whether a set of epsilon values, the difference between each iteration of a set of variables, has been met. If the change in values are less than the epsilon value, ensuring that the statistically sound results have been met, the algorithm proceeds onto the next step. As I-VDT-HMM employs the Viterbi algorithm [3] which requires the iteration over all states, the algorithm is an off-line classification algorithm. The pseudocode for I-VDT-HMM can be found in Algorithms 1, and 2.

HMMs use three probabilities to determine hidden states, the observed probability, transitional probabilities between states, and the probability that any of the previous states will lead to the current state: the emission probability. The transitional probability is calculated using the total number of eye records of a given state over the total number of all eye records. The calculation for transition probabilities can be found in Equation 1, where s is the current state, p is the previous state, and n is the number of eye records. To calculate the observational probabilities, the probability density function (PDF) using the means and standard deviations of the respective feature classifier are used where the results are then

normalized. The equation for the PDF can be found in Equation 2 where σ^2 is the standard deviation, μ is the mean, and x is the current observation. The emission probability for each state is calculated using the product of the previous emission probabilities, the transitional probabilities given the current and previous states, and the observational probability given the current state. The calculation for the emission probability can be found in Equation 3, where s is the current state, p is the previous state, and i is the current state's index.

$$P_{transition,s,p} = \frac{1}{n}\sum_{i=1}^{n}X; X = \begin{cases} 1, & \text{if } p = True \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$P_{observation} = \frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad (2)$$

$$P_{emission,s,p} = P_{emission,p,i-1} * P_{transition,p,s} * P_{observation,s} \quad (3)$$

When attempting to calculate the probability of a fixation given a saccade input, the PDF function used to calculate the observation probability would occasionally return zero. Being that the Viterbi algorithm builds its probabilities off of the previous probability, and that any number multiplied by zero returns zero, this can lead to poor results. Under this circumstance we reset the probability to a lower bound of one ten-thousandth of the other state. In an instance that both probabilities equaled zero, a flag was set for the reset of both probabilities to an initial state. In other cases, the PDF function would return a very small probability. After several iterations this would lead to a cascading effect where one classification would much outweigh the other using the same multiplicative probability property described above. In the instance the lower bound on probabilities has been exceeded, a flag was set to multiply both numbers by the log base 10 of the maximal probability, thus maintaining the probabilistic ratio between either state.

3 METHODOLOGY

3.1 Data

The data used in our experiments was recorded using an EyeLink 1000 eyetracker recorded at 1000 Hz [10] on a 21-in CRT monitor with a refresh rate of 80 Hz and a resolution of 1,024 by 768 pixels. The data consists of 11 subjects whom were recorded in monocular mode and produced various amounts of noise. The stimulus presented to the subjects was a 2-D step-ramp stimulus where the recorded data was converted into degrees of visual angle. This data was originally recorded and published in [7], where you can find additional information.

3.1.1 Data Subsampling. As a high quality eye tracker is not always available, the data was sub-sampled into 7 different sampling frequencies: 30, 50, 60, 100, 200, 300, and 500 Hz. The subject's eye records were then classified into two groups: noisy and clean data. These identifications have been made with the assistance of [7]. The three algorithms evaluated in this paper are tested on 8 of the 11 subjects at each of the 8 sampling frequencies. We will focus primarily on the low, 30 Hz, and high, 1000 Hz, frequencies for clean, subject 007, and noisy, subject 10, datasets for analysis of the reported algorithms.

Input : array of eye position points, velocity threshold - V_t, dispersion threshold - D_t, minimum temporal window size - W_t

Output: list of fixations, saccades, and smooth pursuits

Calculate velocities for each point;

Mark all points above V_t as saccades;

Viterbi Algorithm(array of eye positions, [fixations, saccades], velocity, V_t);

Filter saccades;

Initialize temporal window of size W_t over remaining eye movements;

while *temporal window not reaching end of array* **do**

    Calculate dispersion of points in window;

    if *dispersion < D_t* **then**

        while *dispersion < D_t* **do**

            Mark point as fixation;

            Add unclassified point to window;

            Calculate dispersion of points in window;

        end

    else

        Mark first point as smooth pursuit;

        Move window to next point;

    end

end

Viterbi Algorithm(array of eye positions, [fixations, smooth pursuits], dispersion, [D_t, W_t]);

Merge saccades back into eye position data;

Return list of classifications;

Algorithm 1: I-VDT-HMM

3.2 Behavioral Scores

We use the behavioral scores proposed in [6] and [7] in order to evaluate our algorithm due to their effectiveness in providing meaningful classifications in an automated setting. The behavioral scores provide a better result than direct classification accuracy as they take into account multiple factors found within a healthy individual's HOS which would affect such a classification method. For example, when considering a pursuits quantitative score (PQnS), the latency between seeing the target move and the HOS's response, the pursuit latency, as well as the time it takes for the HOS to catch up to the object are considered. The assumption then is made that the behavior of eye movements is matched with those of a healthy person's HOS. The scores measured in this work are the fixation quantitative score (FQnS), the saccade quantitative score (SQnS), pursuit quantitative score (PQnS), misclassified fixation score (MisFix), fixation qualitative score (PQlS), and the pursuit qualitative scores for positional (PQlS_P) and velocity (PQlS_V) accuracy. The ideal scores for each of the behavioral scores can be found in Table 1.

The FQnS gives the means of computing the amount of fixation points classified correctly as fixations. We use the derivation of the ideal FQnS found in [7], which takes into account the effects of SP on fixation classification. Through our calculations we found 81.5991 to be the ideal behavioral score which differs from the 83.87 score found in [7].

Input : array of eye position points, classes, feature, thresholds
Output : list of classified classes
Calculate mean of feature for classes;
Calculate standard deviation of feature for classes;
Count transitions between each state;
while *not converged* **do**
  Initialize emission and classification matrices;
  Calculate observation probabilities;
  Insert observation probabilities into first column of emission matrix;
  for *column in emission matrix > 1* **do**
    Calculate observation probabilities;
    if *observation probability == 0* **then**
      Set observation probability to lower bound;
    end
    Calculate transitional probabilities;
    Set classification matrix state to highest probability;
    if *emission probability < lower bound* **then**
      Normalize probabilities;
    end
  end
  Calculate maximum final probability;
  for *last record to first record* **do**
    Set previous records highest probability as classification for current record;
  end
  Calculate means and standard deviations of features;
  Count state transitions;
  if *Previous iterations epsilon values are met* **then**
    converged;
  end
end

Algorithm 2: Viterbi Algorithm

The SQnS is defined as the ratio between the detected number of saccades and the total number of saccades in the stimulus [6]. In [7] the SQnS was modified to account for the detection of SPs in the stimulus using a temporal window over the step stimulus. This method considers anticipatory and corrective saccades. Ideally, the SQnS would be 100 as it would indicate the algorithm was able to successfully identify all saccades within the stimulus window.

The intuitive idea behind the PQnS is the ratio of correctly classified SPs over the total amount of SP stimulus. The ideal PQnS takes into account the latency and resulting corrective saccade exhibited by the HOS when given a SP stimulus. The calculation for ideal PQnS may be found in [7]. Our calculations on the ideal score differed slightly from [7], where we calculated it to be 52.03 from their 52.04.

The MisFix score is calculated as the amount of fixation points that are classified as a smooth pursuits over the total amount of fixation stimulus [7]. The ideal score takes into consideration the fact that the termination phase of a SP happens after the SP into the following fixation stimulus. This makes the assumption that

Behavioral Score	Ideal Score
FQnS	81.5991
SQnS	100
PQnS	52.04
MisFix	7.1
PQlS	0
PQlS_P	0
PQlS_V	0

Table 1: Ideal Behavioral Scores

each SP is followed by a fixation. The calculation for the ideal score can be found in [7], which comes out to 7.1.

Intuitively, the FQlS is measured as the euclidean distance between a fixation stimulus and the detected fixation centroid [6]. The score is normalized by the total amount of points compared. The ideal score for FQlS is 0, however, the score is unlikely to be achieved due to inaccuracy of eye trackers and normal behavior of the HOS [6].

The pursuit qualitative scores, PQlS_P and PQlS_V, like the FQlS, are used to compare the proximity as well as the velocity of the detected SPs to the corresponding stimulus [7]. The scores are then normalized over the amount of points compared. Due to calibration errors, corrective saccadic behaviors, HOS latency, and classification errors, the ideal scores of 0 may not be achievable [7].

3.3 Parameter estimation

A parameter estimation algorithm was used in order to detect the ideal velocity and dispersion thresholds for the proposed algorithm. Every 5 velocity thresholds between 70 and 150 were used to optimize saccade detection during the initial phase of our algorithm. Testing every 0.1° between 0.1° and 2.0°'s were used to optimize our dispersion threshold for the later part of our algorithm. As the minimum pause time of the eye is 200 ms [11], and that the minimum amount of time for the HOS to pick up any information is 100 ms [14], the assumption was made that while there is some variability in the HOS, at least 150 ms must pass while fixated in order to detect anything. Thus, a duration threshold of 150ms was employed. As PQlS_P and PQlS_V are generally quite high due to catch up saccadic behavior, a multiplicative weight of 10 was added on to the difference between the classification and the ideal scores for the remaining behavioral scores.

4 RESULTS

4.1 I-BDT

I-BDT was implemented in our environment on our data using the original code linked in [13]. The classification results on clean 1000 Hz and 30 Hz data are shown in Figures 1 and 2 respectively. The assumption made in [13] was that fixations had a velocity of 0, and that the temporal window would be able to correctly discern between smooth pursuits and fixations. As eye trackers and the HOS have inherent noise associated with them, we tested the same 1000 Hz and 30 Hz data on a series of different fixation thresholds. The results are shown in Figures 3 and 4 respectively. These results indicated that the fixation mean, μ, may prove to provide better

results on an average case when considering sample rates. The results of a fixation threshold of μ is shown on clean 1000 Hz data in Figure 5. Using μ as the fixation threshold, we were able to achieve the following results on clean and noisy data on sampling frequencies 30 Hz, 100 Hz, 500 Hz, and 1000 Hz shown in Figure 6.

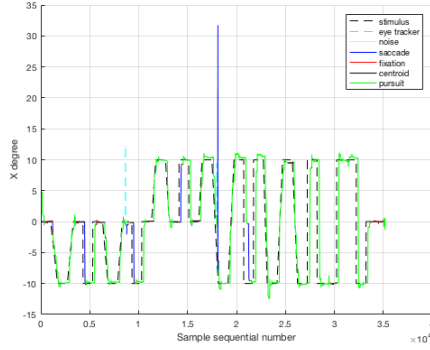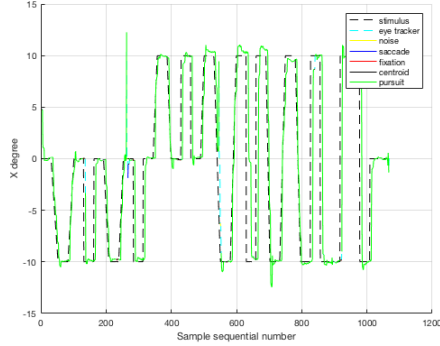**Figure 1: Original I-BDT Code implemented in our environment and ran on subject s-007 on original 1000 Hz data**

Figure 2: Original I-BDT Code Implemented and Ran on Subject s-007 on Sampled 30 Hz Data

4.2 I-VDT

I-VDT was first proposed using the same environment and data we are using [7]. One difference is that we have sampled the data to get an idea of how each algorithm is affected by sampling frequencies. The results of I-VDT on a clean dataset sampled at 1000 Hz is shown in Figure 7. Using the optimal thresholds detailed in [7], we compiled the results on sampling frequencies of 30 Hz, 100 Hz, 500 Hz, and 1000 Hz on both clean and noisy data shown in Figure 8.

4.3 I-VDT-HMM

The proposed algorithm in this work, I-VDT-HMM, was tested on all of our subjects and frequencies in order to determine its optimal thresholds. The results which are averaged across the subjects are shown in Figure 9. The optimal thresholds selected for our data

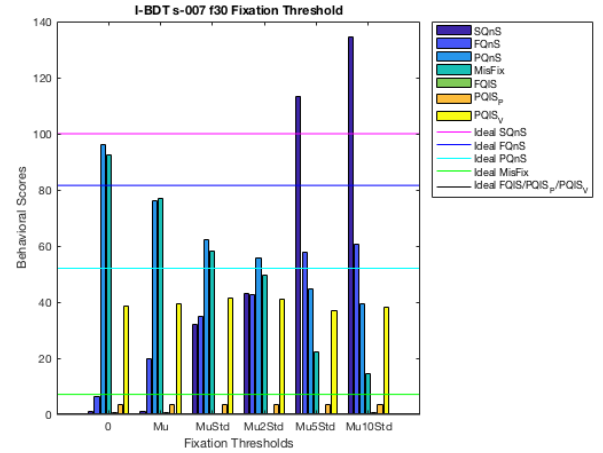**Figure 3: I-BDT Results Using Multiple Fixation Thresholds on 1000 Hz Data**
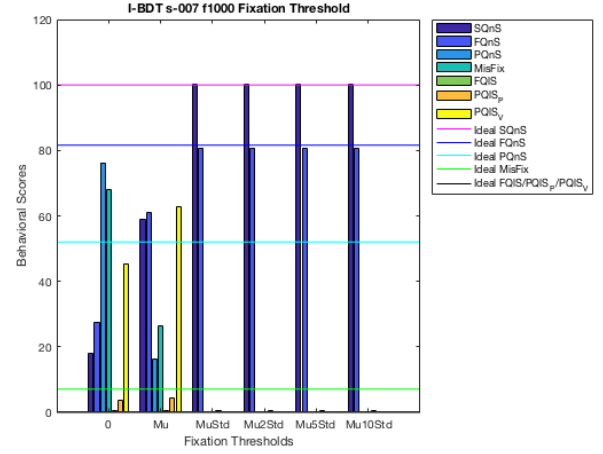
Figure 4: I-BDT Results Using Multiple Fixation Thresholds on 30 Hz Data

| Velocity Threshold	75
Dispersion Threshold	0.67
Duration Threshold	150

Table 2: Optimal Thresholds for I-VDT-HMM

are shown in Table 2. These optimal thresholds give the behavioral scores as shown in Figure 10 on subjects 007 and 010 across frequencies 30 Hz, 100 Hz, 500 Hz, and 1000 Hz. When using these thresholds on eye record data, we achieve the classification results using clean, subject 007, data on 1000 Hz and 30 Hz sampling frequencies shown in Figures 11 and 12 respectively. The behavioral scores attributed to these classification results can be found in Table 3.

Figure 5: I-BDT Results Using μ as Fixation Threshold on 1000 Hz Data
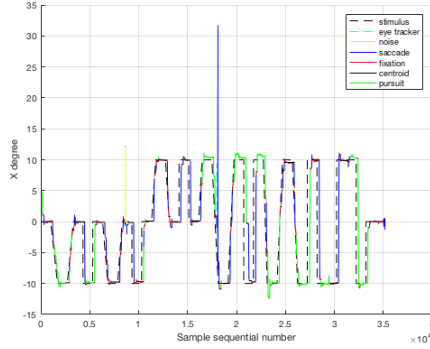
Figure 6: I-BDT Results Using μ as Fixation Threshold Shown in Behavioral Scores on Clean and Noisy 30 Hz, 100 Hz, 500 Hz and 1000 Hz Data
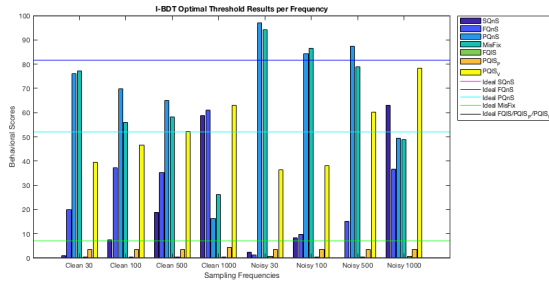
Figure 7: I-VDT Results on Subject 007 Sampled at 1000 Hz
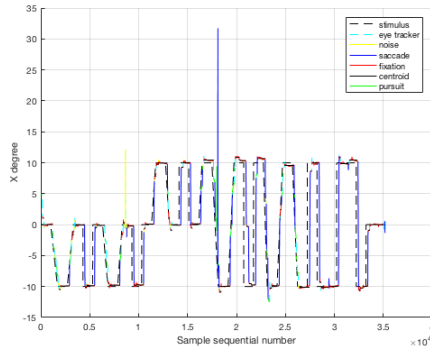
4.4 Cross Algorithm Results

Figure 13 presents the average classification time between algorithms over all tested frequencies and subjects. The X axis represents the frequency the eye records were sampled to, and the Y axis represents the classification time. Figures 14 and 15 compare all algorithms using the clean data set, provided by subject 007, while 16 and 17 compare all algorithms using the noisy data set,

Figure 8: I-VDT Results Using Optimal Thresholds Shown in Behavioral Scores on Clean and Noisy 30 Hz, 100 Hz, 500 Hz and 1000 Hz Data
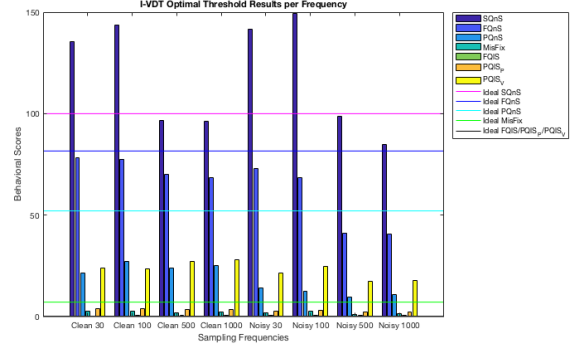
Behavioral Score	1000 Hz Result	30 Hz Result
FQnS	73.91	69.41
SQnS	92.13	102.06
PQnS	48.26	12.43
MisFix	10.01	3.82
FQlS	0.39	0.43
PQlS_P	3.06	6.02
PQlS_V	29.34	86.00

Table 3: Behavioral Scores using I-VDT-HMM for Subject 007 on 1000 Hz and 30 Hz Data

Figure 9: I-VDT-HMM Optimal Thresholds Across Frequencies, Averaged Over Subjects
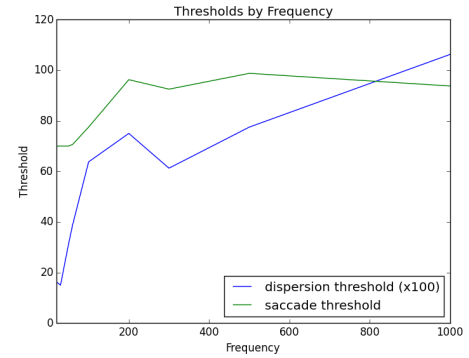
provided by subject 010, on sample frequencies 1000 Hz and 30 Hz respectively.

5 DISCUSSION

On a clean dataset, as shown in the original paper [13], I-BDT was able to achieve outstanding scores averaging 94.98% classification accuracy. On the basis of Figures 1 and 2, we can determine that the assumption made about fixation velocities being equal to zero to be

Figure 10: I-VDT-HMM Optimal Threshold Results given in Behavioral Scores on Clean and Noisy 30 Hz, 100 Hz, 500 Hz, and 1000 Hz Data
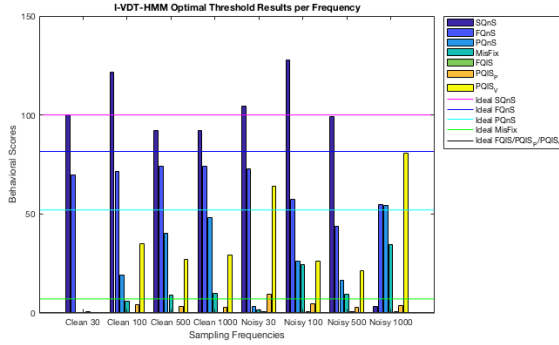

Figure 11: I-VDT-HMM Optimal Threshold Classification Results on Subject 007 at 1000 Hz
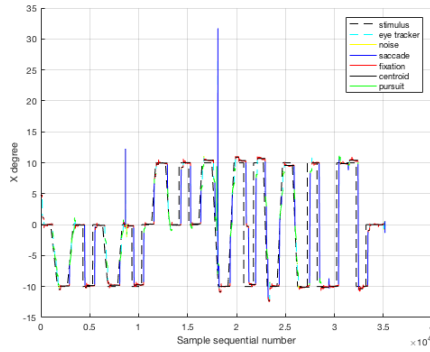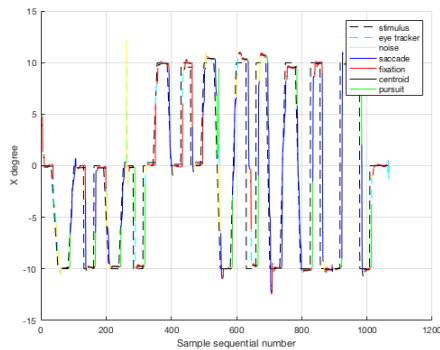

Figure 12: I-VDT-HMM Optimal Threshold Classification Results on Subject 007 at 30 Hz

poorly made when introduced to a relatively noisy dataset. When determining a more optimal fixation threshold, we tested fixation thresholds 0, μ, and several μ's plus multiples of the standard deviation to see how the scores would fair. For lower sampling rates,

Figure 13: Comparison of Algorithms Mean Classification Times by Frequency
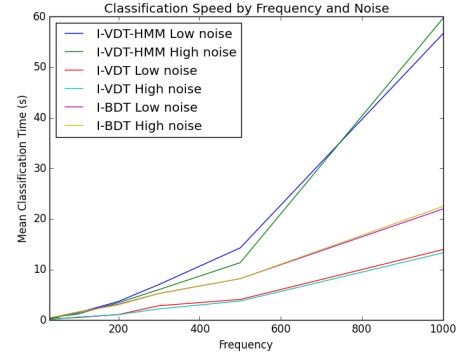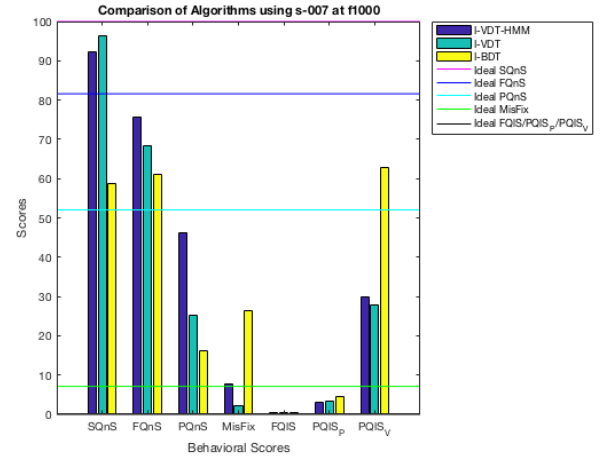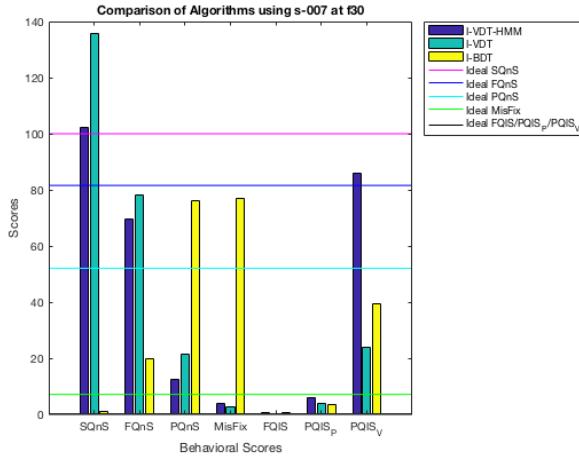

Figure 14: Comparison of Algorithms on Subject 007 Data using 1000 Hz Sample Frequency


the higher the fixation threshold the better the algorithm did. This is likely a result of how noisy data can affect the velocity of points. For higher sampling rates after averaging the overall thresholds we found that the mean velocity of a fixation proved to be the most reliable. Using the mean velocity for the fixation threshold we were able to produce the classification results for clean 1000 Hz data found in Figure 5.

Through the algorithm comparison figures, 14, 15, 16 and 17, we can see that I-VDT and I-VDT-HMM are the most closely matched algorithms, with I-VDT-HMM obtaining a closer-to-ideal score roughly half of the time. However, on the clean 1000 Hz data sets, we obtain significantly more optimal results. Additionally, in Figure 13 we can see that I-VDT is substantially faster than I-VDT-HMM. Considering the motive behind I-VDT-HMM is to use the idea of thresholds to create classifications and then use probabilities to optimize those results, the results make quite a bit of sense. I-VDT-HMM provides the ability to optimize results at the cost of time.

Figure 15: Comparison of Algorithms on Subject 007 Data using 30 Hz Sample Frequency

Figure 16: Comparison of Algorithms on Subject 010 Data using 1000 Hz Sample Frequency
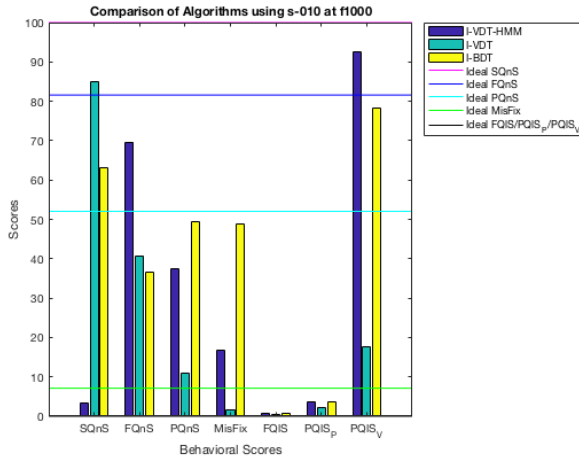
Noticing Figure 16 we can see that the SQnS score is quite low on the high frequency noisy data for I-VDT-HMM. An issue with the Viterbi Algorithm [3] convergence method is that when introduced to data that has too much noise, the noise will greatly affect the mean velocity of each of the states. Being that the mean velocity is included in our epsilon values, adding any additional classification to any state will affect the mean. Assuming the fixation velocity starts out high, as to be expected in noisy data, the mean velocity of saccades will continue to rise quite high until the only saccades left are near peak velocity. Once the mean becomes too high, the PDF function will eventually return a 0 result leading to our described probability normalization functions. It is interesting to note that when this data is subsampled, the scores become much better as can be seen in Figure 17. This is likely due to some of the noise being filtered out, leaving a more clean eye record to classify.

Figure 17: Comparison of Algorithms on Subject 010 Data using 30 Hz Sample Frequency
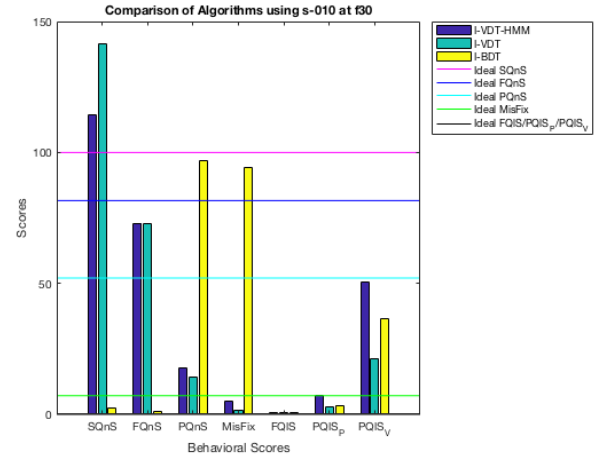
6 CONCLUSION

In this work we provide an analysis of multiple tertiary eye movement classification algorithms: I-VDT [7], a threshold based algorithm, I-BDT [13], a probability based algorithm, and a newly proposed algorithm: I-VDT-HMM, a hybrid threshold and probability based algorithm. In this paper we use behavioral scores introduced in [6] and [7] to assess the quality of each algorithm through the algorithm's classification of normal behavior associated with the HOS. These behavioral scores also led to an automated threshold selection allowing the optimization of the algorithm's results. Our findings show that the proposed algorithm, I-VDT-HMM, significantly outperforms the other two algorithms on high frequency low noise data at the trade off of time.

In the future, to better account for noise, we would like to modify the averages at which we are producing the features for I-VDT-HMM's Viterbi algorithm in the probability distribution function. Additionally, modifying the epsilon functions to detect an ever increasing mean could lead to much more favorable results in high noise samples.

REFERENCES

[1] Raymond Dodge and Thomas Sparks Cline. 1901. The angle velocity of eye movements. *Psychological Review* 8, 2 (1901), 145.

[2] Andrew Duchowski. [n. d.]. T.,(2003),âĂIJEye tracking methodology: Theory and practiceâĂİ. ([n. d.]).

[3] G David Forney. 1973. The viterbi algorithm. *Proc. IEEE* 61, 3 (1973), 268–278.

[4] Elina Isotalo, Aalto Heikki, and Pyykkö Ilmari. 2009. Oculomotor findings mimicking a cerebellar disorder and postural control in severe Meniere's disease. *Auris Nasus Larynx* 36, 1 (2009), 36–41.

[5] Howell Istance, Aulikki Hyrskykari, Lauri Immonen, Santtu Mansikkamaa, and Stephen Vickers. 2010. Designing gaze gestures for gaming: an investigation of performance. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM, 323–330.

[6] Oleg V Komogortsev, Denise V Gobert, Sampath Jayarathna, Do Hyong Koh, and Sandeep M Gowda. 2010. Standardization of automated analyses of oculomotor fixation and saccadic behaviors. *IEEE Transactions on Biomedical Engineering* 57, 11 (2010), 2635–2645.

[7] Oleg V. Komogortsev and Alex Karpov. 2013. Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods* 45, 1 (01 Mar 2013), 203–215. https://doi.org/10.3758/s13428-012-0234-9

[8] R John Leigh and David S Zee. 2015. *The neurology of eye movements*. Vol. 90. Oxford University Press, USA.

[9] A Poole. 2005. LJ Ball Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects. Chapter in C. Chaoui (Ed.): Encyclopedia of HCI. Pennsylvania: Idea Group. (2005).

[10] SR Research. 2010. EyeLink 1000. (2010). http://www.sr-research.com/EL_1000.html

[11] Timothy A Salthouse and Cecil L Ellis. 1980. Determinants of eye-fixation duration. *The American journal of psychology* (1980), 207–234.

[12] Dario D. Salvucci and Joseph H. Goldberg. 2000. Identifying Fixations and Saccades in Eye-tracking Protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (ETRA '00)*. ACM, New York, NY, USA, 71–78. https://doi.org/10.1145/355017.355028

[13] Thiago Santini, Wolfgang Fuhl, Thomas Kübler, and Enkelejda Kasneci. 2016. Bayesian Identification of Fixations, Saccades, and Smooth Pursuits. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 163–170. https://doi.org/10.1145/2857491.2857512

[14] Heino Widdel. 1984. Operational problems in analysing eye movements. *Advances in psychology* 22 (1984), 21–29.