To appear in Proceedings of ACM Conference on Human Factors in Computing Systems (CHI), Atlanta, GA, 2010

Real Time Eye Movement Identification Protocol

Do Hyong Koh
Department of Computer Science
Texas State University-San Marcos
San Marcos, TX 78666 USA
dk1132@txstate.edu

Sandeep Munikrishne Gowda
Department of Computer Science
Texas State University-San Marcos
San Marcos, TX 78666 USA
sm1499@txstate.edu

Oleg V. Komogortsev
Department of Computer Science
Texas State University-San Marcos
San Marcos, TX 78666 USA
ok11@txstate.edu

Abstract
This paper introduces a Real Time Eye Movement Identification (REMI) protocol designed to address challenges related to the implementation of the eye-gaze guided computer interfaces. The REMI protocol provides the framework for 1) eye position data processing such as noise removal, smoothing, prediction and handling of invalid positional samples 2) real time eye movement identification into the basic eye movement types 3) mapping of the classified eye movement data to interface actions such as object selection.

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

Keywords
eye movement, identification, classification, eye tracking, human computer interaction, real time.

ACM Classification Keywords
I.5.2 Pattern Recognition: Design Methodology-Classifier design and evaluation; I.5.5 Pattern Recognition: Implementation-Interactive systems.

General Terms
Algorithms, Design, Performance

Introduction
An eye-gaze-guided computer interface provides an important alternative interaction modality for disabled users or an additional communication channel for normal users [9]. Such interfaces are driven by an eye tracker - equipment that provides the coordinates of the user's eye-gaze on a computer screen.

There are several challenges associated with design and development of such interface that can be divided into three categories. **Processing:** the eye positional data provided by the eye tracker is frequently missing, inaccurate, noisy or/and delayed due to the transmission and processing delays [1,4] **Classification:** raw eye positional data must be

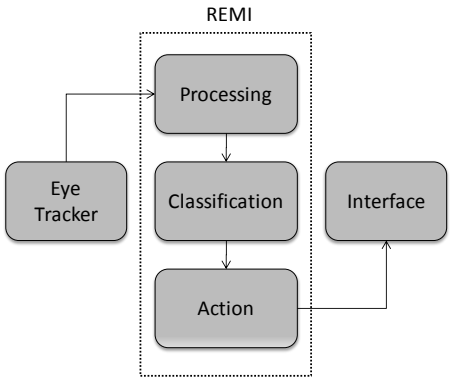

figure 1. Overview of the REMI protocol

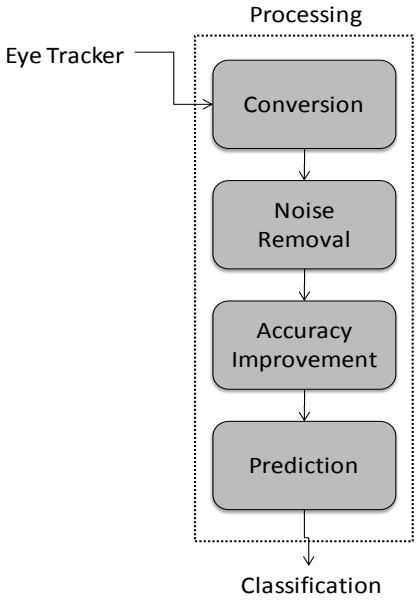

figure 2. Overview of the Processing module

classified by an eye movement identification algorithm in real-time into basic eye movement types (fixations, saccades, pursuits etc.) to provide necessary information for the future interface actions [4,9]. Classification of various eye movement types is not an easy task and provides a formidable challenge even in cases of the off-line classification [8]. **Action:** creating communication tokens from the classified eye movement data and mapping those tokens into specific interface actions [4].

To the best of our knowledge the integration of processing, classification and action into a single reliable protocol was not done before. The goal of the Real Time Eye Movement Identification (REMI) protocol is to provide such integration. In this paper we provide theoretical design and implementation details of the REMI protocol and report results from the application of REMI to a real-time eye-gaze-guided interaction system.

REMI Protocol Design Overview

Figure 1 provides an overview of the REMI protocol. The general goal of the REMI protocol is to process the raw eye positional data and to provide enough information to an interface for the interface to take a specific action. This goal is achieved in three phases Processing, Classification and Action that are described next.

Processing

Very frequently the data received from an eye tracker is provided in the eye tracker units that are hardware/software dependent. The first goal of the processing component is to convert an eye-gaze's coordinates from an eye tracker's units into degrees. The rotation of the eye globe estimated in degrees allows classification of an individual eye gaze position sample into a basic eye movement type by the classification module. The quality of the eye movement data can be affected by the eye blinking, moisture and lighting conditions, therefore producing invalid eye position samples. The second goal is to disregard the invalid or noisy eye position samples substituting them with a carefully selected estimation. The third goal is to improve positional accuracy of the signal. The forth goal is to substitute current eye-gaze coordinates to the predicted coordinates. Such prediction may be necessary in cases of high transmission or/and processing delays and should be applied to gaze-contingent compression systems [5] or systems designed for the instantaneous interaction [7]. Figure 1 provides the overview of the Classification module.

Classification

The goal of this module is to classify the processed eye position samples into specific eye movement types, e.g. fixation (stationary eye movement that provides high acuity picture to the brain), saccade (rapid eye rotation that moves an eye from one fixation to the next), pursuit (smooth eye transition that follows a moving object), etc. [1]. Basic eye movement types represent greatly reduced information space, where each type of movement is described by a set of parameters, e.g., fixation by location and duration, saccade by location, amplitude and direction, pursuit by location, duration, direction and velocity. The design of classification module should provide an easy way to employ any real-time capable classification model.

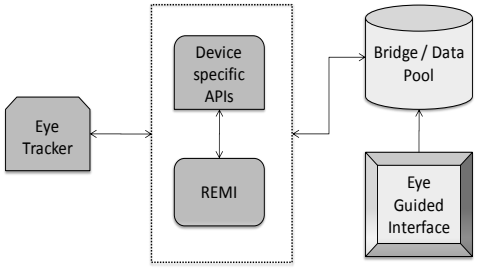

figure 3. REMI implementation diagram

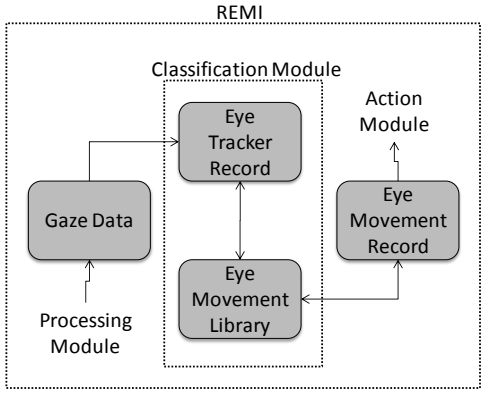

figure 4. Classification module implementation

Action

The goal of the action module is to provide an interface with a history of previous eye movement types and their parameters. This information can be treated by the interface as a sequence of interaction tokens allowing to take a specific action, e.g., fixation duration and location token determines "click" characteristics [9], saccade onset/offset tokens determine "click" parameters [3,7], sequence of fixation and saccade tokens determines a letter typed into a word editor [10]. In addition, the action module can perform post-filtering of the classification module data to discard not needed interaction tokens.

REMI Protocol Implementation

Figure 3 provides an overview of the implementation.

Processing

Device specific application programming interface (API) is highly dependent on the type of the eye tracker used. The API provides eye positional data in the form of a 4-tuple (x,y,t,v), where x and y are coordinates of the eye gaze location is measured in the eye tracker's specific units that very often vary vendor to vendor, t is the time when the sample was taken, v is the validity of the sample that represents the eye tracker's confidence in the accuracy of the sample provided. Conversion of the eye tracker units into degrees is described in [1]. Invalid eye position samples are detected through the parameter v. Noise filtering, smoothing and more accurate estimation of an eye gaze's coordinates is done by a two state Kalman filter [4]. Positional accuracy can be also improved by applying the data collected during eye tracker's calibration procedure. In case when eye gaze prediction is necessary, it can done by a six state Kalman filter [6]. The result of the Processing stage is stored in the Gaze Data record and is sent to the classification module at the eye tracker's sampling frequency.

Classification

Figure 4 provides an overview of the implementation. The Gaze Data information is stored in a queue for further processing. Necessary queue size is determined by a specific eye movement classification model. Velocity based classification models such as Velocity Threshold Identification (I-VT), Hidden Markov Model Identification (I-HMM), Kalman Filter Identification (I-KF) [8] require a queue size of two to conduct velocity calculation. Dispersion based models such as Dispersion Threshold Identification (I-DT), Minimum Spanning Tree Identification (I-MST) [8] require a queue size that is proportional to their temporal sampling window. Acceleration based models such as Finite Input Response Filter (I-FIR) [1] require a queue size that is proportional to the maximum expected saccade's amplitude. It should be noted that larger queue size results in an increased processing time therefore reducing overall system's performance.

Information in the queue is employed in populating of the Eye Tracker Record (ETR) which contains eye positional sample coordinates, velocity, acceleration and other auxiliary data. The Eye Movement Library contains an implementation of the specific classification model such as the I-VT, I-HMM, I-KF, I-DT, I-MST, I-FIR and allows to classify an individual eye-gaze point into a specific basic eye movement type based on the model selected.

A classified eye position sample is employed in the population of the Eye Movement Record (EMR) queue,

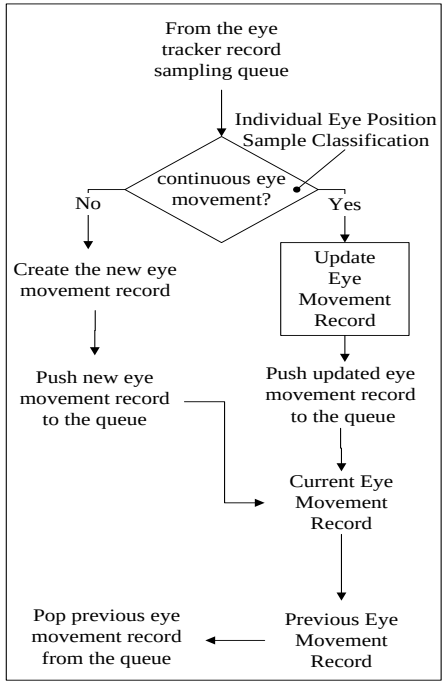

figure 5. The eye movement record generation process in the REMI

which contains an information about previous and current eye movement types experienced by a user. The EMR queue is passed as an input to the Action module. Figure 5 shows the process of the EMR generation or an update by an eye movement classification model. Specifically, the information in the EMR queue is updated or created with each upcoming classified eye position sample (the actual classification of an individual sample is beyond the scope of this paper and can be found through [8]). The update is based on the properties of the newly processed ETR, determining whether it is a part of a continuing or a new movement type. For example, the classification module has to determine if an incoming ETR sample is a part of the current fixation or a first sample of a new eye movement type, if the EMR already contains a detected fixation. In case if the incoming sample is a part of the continuing fixation, fixation properties such as the coordinates and the duration are updated. If the incoming sample is not a part of a fixation, e.g., it is a part of a saccade, a new EMR is created and pushed to the EMR sampling queue. Algorithm which is responsible for the update of the EMR is presented by the Figure 6.

Action

The action module takes the information supplied by the EMR queue to provide a history of the previously detected eye movement types and their parameters which serve as basic communication tokens to an interface.

After every update the EMR queue is passed to the bridge object and the "updated" flag is set. The goal of the bridge/data pool is to provide a communication channel between the REMI and an eye guided interface, making their implementation and the run environments independent of each other. The interface, keeps a check on the "updated" flag and in case when the flag is set, the interface reads the data from the bridge. After this operation the flag is unset. Such algorithm ensures that the interface actions are performed only once, after the data is read. The EMR queue is referenced by the both the bridge and the interface. The REMI has a *write only* access to the bridge object. On the other hand, the interface has *read only* access to the bridge object with an exception to the "update" flag.

The designer of the eye gaze interaction system should decide upon specific interface actions based on the available interaction tokens.

REMI Protocol Practical Results

The REMI was implemented as a part of the development of the iGaze system [2], which is a real-time eye-gaze driven photo viewing application. The iGaze employs simple communication tokens created from fixations, i.e., coordinates of the fixation determine coordinates of the "click" and fixation duration determines the time of the "click". Figure 7 presents a snapshot of the iGaze application.

The REMI evaluation test consisted of 21 volunteers, age 18-25 (average 22.3), with normal or corrected-to-normal vision. The experiments were conducted with a Tobii x120 eye tracker, 24-inch flat panel screen with a resolution of 1980x1200 pix. Subjects were seated approximately 710 mm from the eye tracker. Sampling of the eye position coordinates was done at 120Hz.

```
Algorithm : Update Eye Movement Record
  Input : Classified Eye Tracker Record
  Output : Updated Eye Movement Record

if Classified Movement Type == Fixation then
  if Previous Movement Type == Fixation then
    Update Previous Fixation EMR ( Position, Duration )
  else if Previous Movement Type == Saccade then
    Update Previous Saccade EMR ( Offset Position, Duration, Amplitude )
    Create New Fixation Eye Movement Record }
  else if Previous Movement Type == Smooth Pursuit then
    Update Previous Smooth Pursuit EMR ( Offset Position, Duration,
Velocity )
    Create New Fixation Eye Movement Record

if Classified Movement Type == Saccade then
  if Previous Movement Type == Saccade then
    Update Previous Saccade EMR ( Duration, Amplitude )
  else if Previous Movement Type == Smooth Pursuit then
    Update Previous Smooth Pursuit EMR ( Offset Position )
    Create New Saccade EMR ( Onset Position )
  else if Previous Movement Type == Fixation then
    Create New Saccade EMR ( Onset Position )

if Classified Movement Type == Smooth Pursuit then
  if Previous Movement Type == Smooth Pursuit then
    Update Previous Smooth Pursuit EMR ( Duration, Direction, Velocity )
  else if Previous Movement Type == Saccade then
    Update Previous Saccade EMR ( Offset Position, Duration, Amplitude )
    Create New Smooth Pursuit EMR ( Onset Position, Direction, Velocity )
  else if Previous Movement Type == Fixation then
    Create New Smooth Pursuit EMR ( Onset Position, Direction, Velocity )
```

figure 6. Algorithm: Update Eye Movement Record

Two eye movement classification models were selected to test the REMI protocol – the Velocity Threshold Identification (I-VT) and the Kalman Filter Identification (I-KF) [8]. In case of the I-VT related test, only the Conversion component was employed in the Processing module. In case of the I-KF, the Noise Removal and the Accuracy Improvements employed a two state Kalman filter [4].

Each participant was asked to complete a sequence of five tasks using the iGaze application. Prior to each task, a subject was presented with an image cropped from one of the pictures stored in the iGaze application. The goal of each task was to find the original picture within three minutes. After each task, completion time was recorded. If a participant failed to find the picture, a "time out" event was recorded. Half of the subjects were assigned to complete the tasks using the I-KF model and the remaining half were assigned to complete the tasks using the I-VT model.

Processing

The results of the tests indicate that the I-KF test setup, where the accuracy improvement was done by a Kalman filter, yielded an improvement of 10% of positional accuracy ($F(1,35067)=168.86, p<0.001$). In addition, the Accuracy Improvement module made the iGaze system more tolerable to the eye tracker's calibration accuracy in cases when the initial accuracy was low, i.e., the completion time for the cases with low accuracy did not increase as it happened in cases without the accuracy improvement [2].

Two state Kalman filter, implemented in the Noise Removal module provided an estimate for the invalid eye position samples. As a result the average completion time for the I-KF scenario was 5% faster than for the I-VT ($F(1,17)=4.86, p<0.04$) scenario. In addition, the Noise Removal module made the iGaze system usable for the cases when data loss was extreme, e.g., more than 80%. For more details please see [2].

Classification & Action

Both the I-VT and the I-KF models were able to classify individual eye position samples in real-time. After a preliminary testing, additional processing capabilities were added to the Action module. Specifically, the interaction tokens created by the micro saccades (involuntary saccades with small amplitudes (<2°) which occur during a fixation) were discarded. Such saccades created multiple fixation tokens that hindered the performance of the iGaze system. Such behavior impacted the I-VT setup to a larger degree than the I-KF setup, due to the lack of the Noise Removal component in the Processing module of the I-VT setup.

Discussion

Two models mentioned in this paper are capable of the real-time eye movement classification into fixation and saccades: the I-VT and the I-KF. The I-VT is the easiest model to implement, while the I-KF is more challenging, but provides noise removal, accuracy improvement and prediction capabilities. The I-HMM is real-time performance capable, but requires probabilistic approach to the eye positional data. The I-DT requires larger processing queue sizes, therefore providing higher interaction latency. Both the I-MST and the I-FIR require larger buffer sizes that are proportional to the maximum saccade's duration. In addition, the I-MST model requires high computational processing power. All models are extremely sensitive

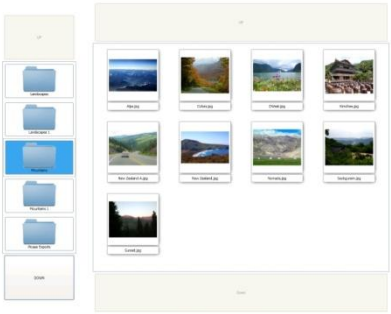

figure 7. iGaze

to the specific values of the input parameters [8]. To the best of our knowledge there are no classification models that can provide tertiary eye movement classification in real-time, i.e., classify eye positional signal into fixations, saccades and pursuits. Such models are needed by the HCI community to provide more versatile interaction experience to the users.

Conclusion

This paper presented the Real-time Eye Movement Identification (REMI) protocol that is created to address challenges associated with the design and the implementation of the real-time eye gaze driven human computer interaction systems making them closer to the practical reality. Specifically three groups of challenges were addressed 1) processing of the missing, noisy, inaccurate, delayed eye positional data 2) classification of the eye positional samples into the basic eye movement types in real-time 3) interface actions based on the classified data. The REMI protocol is hardware/software independent and can be employed with any real-time capable eye movement classification model.

An actual implementation of the REMI protocol with two real-time eye movement classification capable models was reported. The results indicate that the REMI components can be effectively employed for the improvement of the eye positional accuracy, noise removal, real-time eye movement classification and successful interaction with interface components.

Acknowledgements

This work is supported in part by funding from Texas State University - San Marcos to Dr. Komogortsev and Sigma Xi Grant-in-Aid of Research program through grant G200810150639.

References

[1] Duchowski, A. T. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag, London, UK, 2007.

[2] Koh, D., Munikrishne Gowda, S., and Komogortsev, O. Input evaluation of an eye-gaze-guided interface: kalman filter vs. velocity threshold eye movement identification. *In Proc. EICS 2009*, ACM Press (2009), 197-202.

[3] A. Huckauf and M. H. Urbina, "Gazing with pEYEs: towards a universal input for various applications," in Proceedings of the 2008 symposium on Eye tracking research & applications, Savannah, Georgia, 2008, pp. 1-4.

[4] Komogortsev, O. V., and Khan, J. Kalman filtering in the design of eye-gaze-guided computer interfaces. *In Proc. HCII 2007*, (2007), 679–689.

[5] Komogortsev, O. V., and Khan, J. I. Predictive real-time perceptual compression based on eye-gaze-position analysis. *ACM Trans. Multimedia Comput. Commun. Appl.*, (2008), 4, 1-16.

[6] Komogortsev, O. V., and Khan, J. Eye movement prediction by Kalman filter with intergrated linear horizontal oculomotor plant mechanical model. *In Proc. ETRA 2008*, (2008), 229-236.

[7] Komogortsev, O. V., Ryu, Y., Koh, D., and Munikrishne Gowda, S. Instantaneous saccade driven eye gaze interaction. *ACM International Conference on Advances in Computer Entertainment Technology*, (2009).

[8] Komogortsev, O.V., Jayarathna, U. K. S., Koh, D. and Munikrishne Gowda, S. *Qualitative and quantitative scoring and evaluation of the eye movement classification algorithms*. *In Proc. ETRA 2010,* (2010).

[9] Sibert, L. E., and Jacob, R. J. K. Evaluation of eye gaze interaction. *In Proc. SIGCHI conference on Human factors in computing systems 2000*, ACM Press (2000), 281-288.

[10] Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. *In Proc. ETRA 2008,* ACM Press (2008), 11-18.