Department of Computer Science
San Marcos, TX 78666

Report Number TXSTATE-CS-TR-2009-17

Biometric Identification via an Oculomotor Plant Mathematical Model

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

Sampath Jayarathna
Department of Computer Science
Texas State University-San Marcos
sampath@txstate.edu

Cecilia R. Aragon
Computational Research Division
Lawrence Berkeley National Lab
CRAragon@lbl.gov

Mechehoul Mahmoud
Department of Computer Science
Texas State University-San Marcos
mm2026@txstate.edu

2009-11-19

Biometric Identification via an Oculomotor Plant Mathematical Model

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

Sampath Jayarathna
Department of Computer Science
Texas State University-San Marcos
sampath@txstate.edu

Cecilia R. Aragon
Computational Research Division
Lawrence Berkeley National Lab
CRAragon@lbl.gov

Mechehoul Mahmoud
Department of Computer Science
Texas State University-San Marcos
mm2026@txstate.edu

Abstract

There has been increased interest in reliable, non-intrusive methods of biometric identification due to the growing emphasis on security and increasing prevalence of identity theft. This paper presents a new biometric approach that involves an estimation of the unique oculomotor plant (OP) or eye globe muscle parameters from an eye movement trace. These parameters model individual properties of the human eye, including neuronal control signal, series elasticity, length tension, force velocity, and active tension. These properties can be estimated for each extraocular muscle, and have been shown to differ between individuals. We describe the algorithms used in our approach and the results of an experiment with 41 human subjects tracking a jumping dot on a screen. Our results show improvement over existing eye movement biometric identification methods. The technique of using Oculomotor Plant Mathematical Model (OPMM) parameters to model the individual eye provides a number of advantages for biometric identification: it includes both behavioral and physiological human attributes, is difficult to counterfeit, non-intrusive, and could easily be incorporated into existing biometric systems to provide an extra layer of security.

CR Categories: I.6.4 [Simulation and Modeling]: Model Validation and Analysis; J.7 [Computers in Other Systems]: Process control, Real time.

Keywords: biometrics, oculomotor plant, eye tracking.

1 Introduction

Accurate, non-intrusive, and unforgeable identity recognition is an area of increasing concern to just about everyone in today's networked world, with the need for security set against the goals of easy access. The majority of the world's population would like secure access to their assets without risk of identity theft, yet do not want to be subjected to inconvenient or intrusive detection systems. Many of the most-commonly utilized methods for identity determination have known problems. For example, password verification has demonstrated many weaknesses in areas of accuracy (there is no way to verify that the individual typing the password is actually its owner, unless a temporal pattern recognition system is employed [1]), usability (people forget passwords [2]), and security (people write them down or create easy-to-hack passwords [3]) .

As a result, techniques of biometric identification, defined as methods for identifying persons based on uniquely identifying physical or behavioral traits, have been garnering significant recent interest [4-6]. The potential for advancement in biometric identification methods is substantial due to recent improvements in computer processing power, database size, and sensor technologies.

There are a number of methods employed today for biometric purposes. Some examples include the use of fingerprints, iris and retina scans, face recognition, hand/finger geometry, and voice recognition [5-8] .

Current biometric identification technologies are somewhat fraud resistant, but they are not completely foolproof and may be compromised with available technologies. Even though fingerprint identification is a popular methodology, such systems have been demonstrated to be insufficiently invulnerable in high security environments. Several recent studies have shown that it is possible to fool fingerprinting systems with common household articles such as gelatin [9].

Face recognition systems are still undergoing research to improve their precision and recall [6, 10, 11]. Additionally, identical twins (1:10,000 probability), and related issues such as family resemblance may bring the reliability of such systems into question. It is also possible to use still images and video footage of a person to bypass a face recognition system.

Further disadvantages of many of these methods involve the ability to forge replicas – in some cases even causing injury to the owner of the body part used for biometric identification. Eye movements, in contrast, constitute a behavioral characteristic which is extremely difficult to forge, and which cannot be stolen from an individual.

The challenge lies in classifying eye movements in such a manner that the differences between individuals are more significant than changes in a single human's behavior over time. In order to do this, we turn to physical and behavioral characteristics that are relatively constant in an individual human over their lifetime: the physical structure and behavior of the muscles that move the eye. We propose a person identification method based on the Oculomotor Plant Mathematical Model (OPMM) developed by Komogortsev and Khan [12]), derived from earlier work by Bahill [13]. The OPMM models a human eye as a system that consists of an eye globe driven by a set of extraocular muscles. This system models the anatomical structure of the human eye, where each extraocular muscle is driven by a uniquely defined neuronal control signal and consists of series elasticity, length tension, force velocity, and active tension components. The OPMM as a whole also includes passive elasticity, viscosity and inertia. However, the specific values for the previously determined OPMM parameters were obtained from studies that examined a single individual. In this paper, we analyze eye movement traces and derive a unique vector of values corresponding to each person over a sample set of 41 individuals. We report our results and discuss the challenges and advantages provided by biometric identification via an OPMM.

This paper is organized as follows: section 2 provides a high-level description of the overall biometric identification system; sections 3-7 describe the algorithms used and the OPMM, including details of the pseudocode and equations developed; section 8 describes our experimental methodology; and section 9 presents our results. The final section discusses the results and presents our conclusions.

2 Biometric Identification by Oculomotor Plant Mathematical Model

An overview of our method for biometric identification by Oculomotor Plant Mathematical Model (OPMM) is depicted in Figure 1. The recorded eye movement signal u(t) from a single individual is supplied to the "Eye Movement Classification" module that classifies eye position signal into fixations and saccades. We focus on the detected saccade trajectories, represented by Θ(t) in the diagram. The detected saccade parameters, the onset and offset coordinates and amplitudes of the detected saccades, depicted by h(t), are sent to the second module labeled Oculomotor Plant Mathematical Model (OPMM), which generates simulated saccade trajectories represented by the signal x(t). The difference between detected saccade trajectories Θ(t) and

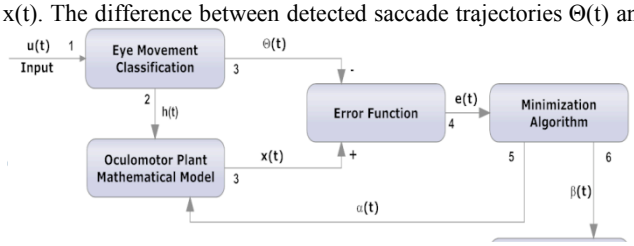

Figure 1. Biometric Identification via an Oculomotor Plant Mathematical Model

simulated saccade trajectories x(t) is computed by the "Error Function" module and the resulting error e(t) is produced. The e(t) signal serves as an input to the "Optimization Algorithm" module that provides feedback to the OPMM module with the goal of minimizing the error e(t). After several iterations, the optimum OPMM parameters in the form of an "Oculomotor Coefficients Vector" module are supplied to the "Person Identification" module which performs the actual identification. Detailed descriptions of each module are provided in the sections to follow.

3 Eye Movement Classification

We employed the Velocity-Threshold (I-VT) algorithm [14] in the Eye Movement Classification module to split an eye movement recording into fixations and saccades. A velocity threshold of 55°/s was employed to separate fixations from saccades. The original I-VT algorithm was modified to output such characteristics of an individual saccade as onset (θ_{x_onset}), offset (θ_{x_offset}) coordinates, amplitude (θ_{sac_amp}) and the coordinates of all eye position points between the onset and the offset. The outputs of this module are the detected saccade trajectories Θ(t) and the onset and offset coordinates and the amplitude of the detected saccades, h(t). Θ(t) is passed as input to the Error Function module described in Section 5, and h(t) is used as input to the OPMM module described in the following section.

4 Oculomotor Plant Mechanical Model

The eye globe rotates in its socket through the use of six muscles. These six muscles are: the medial and the lateral recti – the muscles responsible for horizontal eye movements; superior and inferior recti – the muscles responsible for vertical eye movements; superior and inferior oblique – the muscles responsible for eye rotations around its primary axis of sight; and vertical eye movements. The brain sends a neuronal control signal to each muscle to direct the muscle to perform its work. A neuronal control signal is anatomically implemented as a neuronal discharge that is sent through a nerve to a designated muscle from

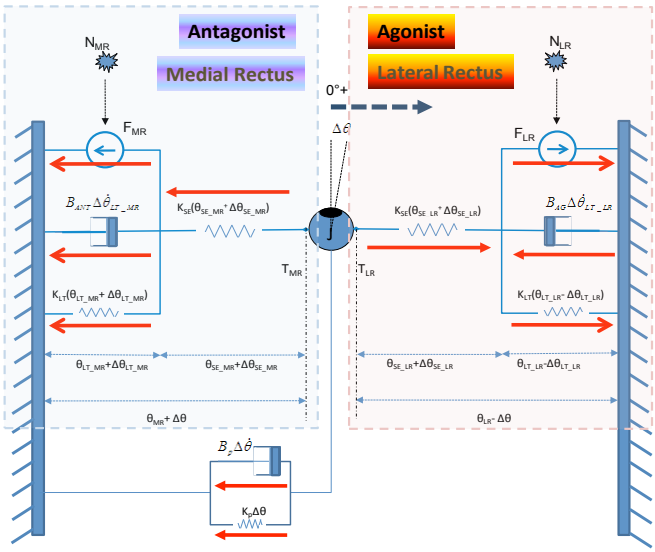

Figure 2. The oculomotor plant mathematical model employed for positive amplitude saccades. Arrow show the direction of forces for each component ΔΘ – eye rotation J is the rotational inertia of the eye globe.

a specific part of the brain [15]. The frequency of this discharge determines the level of muscle innervation and results in a specific amount of work that a muscle can perform. During saccades the neuronal control signal for each muscle resembles a pulse-step function [13]. The eye position during the onset of a saccade, the saccade's amplitude and direction define pulse and step parameters of the control signal. During eye fixations, the neuronal discharge occurs at a constant rate that is linearly related to the eye position.

The OPMM presented in this subsection represents the anatomy of the right eye and presents mathematical equations for rightward saccades; a more detailed description of the model can be found in [12, 16]. The rightward saccades are performed by the lateral rectus (the muscle that is closer to the ear) as the agonist and by the medial rectus (the muscle that is closer to the nose) as the antagonist. The agonist muscle pulls the eye globe in the required direction and the antagonist muscle resists the pull.

The agonist's muscle parameters are identified with a "AG" subscript (example: N_{AG}, B_{AG}), the antagonist with "ANT," the lateral rectus with "LR" and the medial rectus with "MR." Note that either the lateral or the medial recti can play the role of the agonist or the antagonist. Parameters without those subscripts are identical to both types of muscles. The lateral and the medial recti are modeled through a system of mechanical components mimicking the anatomical properties of a muscle.

4.1.1 Muscle Properties

A muscle is a very complex structure [17]. The *Muscle Mechanical Model* (MMM) [12] can be represented via several components. These components are the following: passive elasticity, active state tension, series elasticity, a length-tension component and a force velocity relationship [13, 18, 19].

Passive Elasticity: Each body muscle in the rest state is elastic. The rested muscle can be stretched by applying force. The muscle extension is proportional to the force applied. The passive

elasticity results largely from the meshwork of connective tissues within the muscle, whose fibers become progressively taut when the muscle is stretched [17]. The passive elasticity component is approximated as a linear spring with coefficient K_p=0.5 grams of tension per degree [18].

Active State Tension: Each muscle produces active state tension when stimulated. The frequency of the neuronal discharge is determined by the brain, depending of the type of the eye movement and eye position. When stimulation occurs, a muscle develops tension, trying to contract. The resulting tension is called the active state tension. The intensity of the active state tension depends upon the frequency of the neuronal discharge. An ideal force generator component is used in the MMM to represent active state tension as F_{LR} for the lateral rectus and as F_{MR} for the medial rectus.

Length Tension Relationship: The tension that a muscle develops partially depends on its length. This happens due to the physical structure of the muscles and the propagation of the neuronal control signal in the muscle tissue [17]. In the OPMM model we approximate this relationship as an ideal linear spring with coefficient K_{LE}=1.2grams/º [13].

Series Elasticity: The series elasticity is in series with the active force generator, hence the name. Anatomically, spring elasticity components are located on the tendon and in myosin and in the actin bridges of the muscle fibers. In the MMM, the series elasticity component is modeled as an ideal linear spring with coefficient K_{SE}=2.5grams/º [18].

Force Velocity Relationship: This relationship shows that a muscle is capable of producing larger forces at lower velocities. This dependency of force upon velocity varies for different levels of a neuronal control signal and depends on whether a muscle shortens or stretches. In the OPMM, the velocity of muscle contraction is connected to a change of length in the length tension component of the muscle and represented by the following variables: $\Delta\dot{\theta}_{LT\_LR}$ - lateral rectus, $\Delta\dot{\theta}_{LT\_MR}$ - medial rectus.

4.2 Modeling Saccades

Saccades of positive amplitude are performed by the OPMM presented in Figure 1. The eye rotates by Δθ degrees. Prior to a saccade, the displacement (distance from its equilibrium length) inside the series elasticity component of the lateral rectus is θ_{SE_LR}, and the length tension component is θ_{LT_LR}. Therefore the total displacement inside of the lateral rectus is $\theta_{LR}=\theta_{SE\_LR}+\theta_{LT\_LR}$. Similar logic applies to the medial rectus.

4.2.1 Agonist Muscle Mechanical Model of Lateral Rectus

The right section of Figure 1 shows neuronal firing inducing the contraction of the lateral rectus by generating the active state tension. During contraction, the length tension component shortens by $\Delta\theta_{LT\_LR}$, and the series elasticity component lengthens by $\Delta\theta_{SE\_LR}$, resulting in the eye rotation of Δθ degrees.

The forces that work inside of the agonist MMM during eye rotation can be broken into two force groups. The first group consists of the active state tension F_{LR}; length tension force $K_{LT}(\theta_{LT\_LR}-\Delta\theta_{LT\_LR})$, working in the same direction as F_{LR}; damping force $-B_{AG}\Delta\dot{\theta}_{LT\ LR}$, accounting for the force velocity relationship resisting the contraction of the lateral rectus. Note that the amount of resistive force produced by the damping component is based upon the velocity of contraction of the length tension $\Delta\dot{\theta}_{LT\_LR}$. Summing the forces of the first group we get the following equation:

$$T_{LR} = F_{LR} + K_{LT}(\theta_{LT\_LR} - \Delta\theta_{LT\_LR}) - B_{AG}\Delta\dot{\theta}_{LT\_LR} \quad \mathbf{1}$$

The second force group consists of the series elasticity component that propagates the force T_{LR}, generated by the first group, to the eye globe. The series elasticity component is a linear spring; therefore, T_{LR} can be computed as:

$$T_{LR} = K_{SE}(\theta_{SE\_LR} + \Delta\theta_{SE\_LR}) \quad \mathbf{2}$$

The previous two equations can be rearranged in a form that calculates the force T_{LR} in terms of the eye rotation $\Delta\theta$, and displacement $\Delta\theta_{LT\_LR}$.

$$T_{LR} = \frac{\hat{F}_{LR}K_{SE}}{K_{SE}+K_{LT}} - \frac{\Delta\theta K_{SE}K_{LT}}{K_{SE}+K_{LT}} - \hat{B}_{AG}\Delta\dot{\theta}_{LT\_LR} \quad \mathbf{3}$$

$$T_{LR} = K_{SE}(\Delta\theta_{LT\_LR} - \Delta\theta) \quad \mathbf{4}$$

where $\hat{F}_{LR} = F_{LR} - K_{SE}(\theta_{LR} - \theta_{LT\_LR}) + K_{LT}\theta_{LT\_LR}$.

4.2.2 Antagonist Muscle Mechanical Model of Medial Rectus

The left side of Figure 2 presents the model. The medial rectus is extended by the pull of the lateral rectus. The length tension component extends by $\Delta\theta_{LT\ MR}$ and the series elasticity component extends by $\Delta\theta_{SE\ MR}$.

Two groups of muscle forces similar to the agonist case are present. The first group consists of the active state tension $-F_{MR}$, resisting the agonist pull; length tension force $-K_{LT}(\theta_{LT\_MR} + \Delta\theta_{LT\_MR})$, resisting the agonist pull; damping force $-B_{ANT}\Delta\dot{\theta}_{LT\ MR}$, accounting for the force velocity relationship, resisting the lengthening of the medial rectus. Summing the forces of the first group, we obtain the following equation:

$$T_{MR} = -F_{MR} - K_{LT}(\theta_{LT\_MR} - \Delta\theta_{LT\_MR}) - B_{ANT}\Delta\dot{\theta}_{LT\_MR} \quad \mathbf{5}$$

T_{MR} can be also computed through the properties of the series elasticity component.

$$T_{MR} = -K_{SE}(\theta_{SE\_MR} + \Delta\theta_{SE\_MR}) \quad \mathbf{6}$$

The previous two equations can be rearranged in a form that calculates the force T_{MR} in terms of the eye rotation $\Delta\theta$, and displacement $\Delta\theta_{LT\_MR}$.

$$T_{MR} = -\frac{\hat{F}_{MR}K_{SE}}{K_{SE}+K_{LT}} - \frac{\Delta\theta K_{SE}K_{LT}}{K_{SE}+K_{LT}} - \hat{B}_{AG}\Delta\dot{\theta}_{LT\_MR} \quad \mathbf{7}$$

$$T_{MR} = -K_{SE}(\Delta\theta - \Delta\theta_{LT\_MR}) \quad \mathbf{8}$$

where $\hat{F}_{MR} = F_{MR} - K_{SE}(\theta_{MR} - \theta_{LT\_MR}) + K_{LT}\theta_{LT\_MR}$.

4.2.3 Neuronal Control Signal

Each saccade is generated by a neuronal control signal that looks like a pulse step function [19]. This signal can be presented by the following equations:

$$N_{AG\_sac}(t) = \begin{cases} N_{AG\_sac\_onset}, & t_{sac\_onset} \le t < t_{AG\_sac\_pulse\_onset} \\ N_{AG\_sac\_pulse}, & t_{AG\_sac\_pulse\_onset} \le t < t_{AG\_sac\_pulse\_offset} \\ N_{AG\_sac\_offset}, & t_{AG\_sac\_pulse\_end} \le t < t_{sac\_offset} \end{cases}$$

$$N_{ANT\_sac}(t) = \begin{cases} N_{ANT\_sac\_onset}, & t_{sac\_onset} \le t < t_{ANT\_sac\_pulse\_onset} \\ N_{ANT\_sac\_pulse}, & t_{ANT\_sac\_pulse\_onset} \le t < t_{ANT\_sac\_pulse\_offset} \\ N_{ANT\_sac\_offset}, & t_{ANT\_sac\_pulse\_offset} \le t < t_{sac\_offset} \end{cases}$$

t_{name} constants present time parameters for each type of muscle and action phase. t is the time elapsed from the beginning of the saccade. The OPMM utilized in this paper uses the time constant values presented below.

$t_{sac\_onset} = 0,$

$t_{sac\_offset} = (2.2 * |\theta_{sac\_amp}| + 21)\ ms,$

θ_{sac_amp} is the amplitude of the saccade measured in degrees.

The agonist and antagonist muscle related time constants are:

$t_{AG\_sac\_pulse\_onset} = t_{sac\_onset} + 3\ msec.,$

$t_{AG\_sac\_pulse\_offset} = (t_{AG\_sac\_pulse\_onset} + |\theta_{sac\_amp}| + 10)\ ms$

$t_{ANT\_sac\_pulse\_onset} = t_{sac\_onset},$

$t_{ANT\_sac\_pulse\_offset} = (t_{AG\_sac\_pulse\_onset} + |\theta_{sac\_amp}| + 16)\ ms$

The value representing the width of the antagonist pulse ($t_{ANT\_sac\_pulse\_offset} - t_{ANT\_sac\_pulse\_onset}$) is selected as a result of physiological measurements that indicate that the agonist pulse starts 3ms after the start of the antagonist pulse and ends 3ms before the end of the antagonist pulse.

The parameters such as amplitude, the eye position at the beginning and end of the saccade should be supplied to the OPMM in terms of neuronal control signal. Therefore, we have created a set of functions that transform saccade parameters into neuronal control signal: $N_{AG\_sac\_onset}$, $N_{ANT\_sac\_onset}$, $N_{AG\_sac\_offset}$, $N_{ANT\_sac\_offset}$.

$$N_{AG\_sac\_onset}(\theta_{x\_onset}) = \begin{cases} (20.6 + 2.37|\theta_{x\_onset}|), & if\ agonist\ prior\ to\ saccade \\ 20.6 - 0.74|\theta_{x\_onset}|, & if\ antagonist\ prior\ to\ saccade \end{cases}$$

$$N_{AG\_sac\_pulse}(\theta_{sac\_onset}, \theta_{sac\_amp}) = N_{AG\_fix}(|\theta_{sac\_start} + \theta_{sac\_amp}|) + 190\left(1 - e^{-\frac{|\theta_{sac\_amp}|}{50}}\right)$$

$$N_{AG\_sac\_offset}(\theta_{sac\_end}) = \begin{cases} (20.6 + 2.37|\Delta\theta|), & if\ agonist\ after\ saccade \\ 20.6 - 0.74|\Delta\theta|, & if\ antagonist\ after\ saccade \end{cases}$$

$$N_{ANT\_sac\_onset}(\theta_{x\_onset}) = \begin{cases} N_{AG\_FIX}(\theta_{x\_onset}), if\ agonist\ prior\ to\ saccade \\ N_{ANT\_FIX}(\theta_{x\_onset}), if\ antagonist\ prior\ to\ saccade \end{cases}$$

$$N_{ANT\_sac\_pulse}(\theta_{sac\_amp}) = 0.5$$

$$N_{ANT\_sac\_end}(\theta_{x\_offset}) = \begin{cases} N_{AG\_FIX}(\theta_{x\_offset}), & if\ agonist\ after\ saccade \\ N_{ANT\_FIX}(\theta_{x\_offset}), & if\ antagonist\ after\ saccade \end{cases}$$

θ_{x_onset} and θ_{x_offset} are the horizontal coordinates of the saccade's onset and offset points respectively.

4.2.4 Active State Tension

Though the neuronal control signals $N_{AG\_sac}(t)$ and $N_{ANT\_sac}(t)$, rise and drop instantaneously, neither the forces that muscles apply to the eye globe nor active state tensions rise to their maximum values immediately. This happens due to the anatomical characteristics of the neuronal signaling [13]. In the proposed model, the active state tension is a result of a low pass filtering process performed upon the neuronal control signal. Active state tension dynamics can be represented with the following differential equations at each time interval.

$$\dot{\hat{F}}_{AG}(t) = \frac{N_{AG\_sac} - \hat{F}_{AG}(t)}{\tau_{AG\_sac}} \quad \mathbf{9}$$

$$\dot{\hat{F}}_{ANT}(t) = \frac{N_{ANT\_sac} - \hat{F}_{ANT}(t)}{\tau_{ANT\_sac}} \quad \mathbf{10}$$

τ_{AG_sac} and τ_{ANT_sac} are functions that define the low pass filtering process [13]; they are defined by the activation and deactivation time constants.

4.3 Oculomotor Plant Mechanical Model Equations

The lateral rectus as the agonist applies the force to the eye globe that can be calculated by Equations 3and 4. Those equations can be combined together:

$$K_{SE}(\Delta\theta_{LT\_LR} - \Delta\theta) = \frac{\hat{F}_{LR}K_{SE}}{K_{SE} + K_{LT}} - \frac{\Delta\theta K_{SE}K_{LT}}{K_{SE} + K_{LT}} - \hat{B}_{AG}\Delta\dot{\theta}_{LT\_LR} \quad \mathbf{11}$$

The medial rectus as the antagonist applies the force to the eye globe that can be calculated by Equations 7 and 8. Those equations can be combined together:

$$K_{SE}(\Delta\theta - \Delta\theta_{LT\_MR}) = \frac{\hat{F}_{MR}K_{SE}}{K_{SE} + K_{LT}} + \frac{\Delta\theta K_{SE}K_{LT}}{K_{SE} + K_{LT}} + \hat{B}_{ANT}\Delta\dot{\theta}_{LT\_MR} \quad \mathbf{12}$$

Applying Newton's second law, the sum of all forces acting on the eye globe equals the acceleration of the eye globe multiplied by the inertia of the eye globe.

$$J\Delta\ddot{\theta} = T_{LR} - T_{MR} - K_p\Delta\theta - B_p\Delta\dot{\theta} \quad \mathbf{13}$$

J=0.000043 grams-s^2/degrees - eye globe's inertia, $\Delta\theta$ - eye rotation, $\Delta\dot{\theta}$ velocity of the eye rotation, $\Delta\ddot{\theta}$ eye rotation acceleration, B_p=0.06 grams-s/degrees – viscosity of the tissues around the eye globe. T_{LR} can be calculated by Equation 4 and T_{MR} can be calculated by Equation 8. Thus Equation 13 can be transformed into:

$$J\Delta\ddot{\theta} = K_{SE}(\Delta\theta_{LT\_LR} - \Delta\theta) - K_{SE}(\Delta\theta - \Delta\theta_{LT\_MR}) - K_p\Delta\theta - B_p\Delta\dot{\theta} \quad \mathbf{14}$$

Equations 9, 10, 11, 12, 14 are five differential equations with six variables ($\Delta\theta$, $\Delta\theta_{LT\_LR}$, $\Delta\theta_{LT\_MR}$, $\Delta\dot{\theta}$, $\hat{F}_{LR}$, $\hat{F}_{MR}$). A sixth differential equation can be added as:

$$\Delta\dot{\theta} = \Delta\dot{\theta} \quad \mathbf{15}$$

The solution of these six differential equations will provide an eye movement trajectory $\Delta\theta$ given the value of saccade's amplitude θ_{sac_amp}, the onset θ_{x_onset} and the offset θ_{x_offset} coordinates.

5 Error Function

Saccade trajectories generated by the Eye Movement Classification module and the OPMM module are supplied to the Error Function module, where the error e(t) between those two signals is computed. We compute the Root Mean Squared Error (RMSE) between the detected x(t) and the simulated OPMM $\Delta\theta(t)$ eye position signal.

$$e(t) = RMSE_{1D} = \sqrt{\frac{\sum_{t=t_{onset}}^{t_{offset}} (\mathrm{x(t)} - \Delta\theta(t))^2}{t_{offset} - t_{onset}}} \quad \mathbf{16}$$

When multiple saccades are detected for an individual by the eye movement classification algorithm, the average of the RMSEs from the detected and simulated trajectories is presented as a final e(t). Note that a good approximate solution of the OPMM equations creates an eye movement trajectory with a sampling rate of 1000Hz [16]; in the case where the eye-tracking frequency is lower, the signal $\Delta\theta(t)$ is down-sampled to match the eye tracking frequency.

6 Optimization Algorithm

6.1 Oculomotor Plant Parameters Vector

The goal of the Optimization Algorithm module is to provide a set of better values for the OPMM parameters by minimizing error e(t). The OPMM's parameters such as: passive elasticity, viscosity, series elasticity, length tension, force velocity relationship, height and width on the neuronal control signal are unique for each individual. Some of the values for these parameters were estimated from a record of just one subject [13, 18]. Moreover, some of the parameters such as length tension and series elasticity were derived by manual data fitting and hand-drawn straight line approximations [13]. Parameters related to the innervations of the agonist muscle were adjusted by visually

comparing the output to the trajectory of the actual eye movement [13]. Obviously, the values of the OPMM parameters derived in this way can be improved to provide a much better fit for a specific individual.

One of the ways to derive more accurate values for the OPMM parameters is to employ an optimization algorithm that selects new values for the OPMM parameters with an objective of minimizing the error e(t).

It is important to note that some parameters provide higher impact than other parameters on the simulated eye movement trajectory [13]. The ranking of the parameters starting with those providing the highest influence on the simulated saccade trajectory is as follows: the width of the pulse of the neuronal control signal for the agonist muscle ($t_{AG\_pulse\_width} = t_{AG\_sac\_pulse\_offset} - t_{AG\_sac\_pulse\_onset}$), pulse height of the neuronal control signal for the agonist muscle ($N_{AG\_sac\_pulse}$), length tension (K_{LT}), series elasticity (K_{SE}), passive viscosity of the eye globe (B_p) and force velocity relationship in the agonist muscle represented by the damping component ($\hat{B}_{AG}$), combined passive elasticity of the eye globe and all extraocular muscles (K_p), eye globe inertia (J). All these parameters are selected for actual person identification.

6.2 Optimization Algorithms & Strategies

6.2.1 Optimization Algorithms

We employed two optimization algorithms to determine optimized values for the OPMM parameter vectors ($LR_p = t_{AG\_pulse\_width}$, $LR_s = N_{AG\_sac\_pulse}$, K_{LT}, K_{SE}, $\hat{B}_{AG}$, K_p, J) with an objective of minimizing the error e(t).

First, the Trust-Region (TR) algorithm that uses the interior-reflective Newton method was applied [20]. The TR algorithm is an optimization method that looks for a better value of the input parameter given its initial value. The search for a better value occurs in an area called the trusted region around the initial parameter value. At the start, the region of search is close to the initial parameter value and if a better value is found the trusted region size is increased, otherwise the size of the search region is reduced.

Additionally, the Nelder-Mead (NM) simplex algorithm was applied [21]. This algorithm employs a simplex of n+1 points for a vector y with n dimensions. At the beginning the algorithm builds a simplex around the initial value i by adding a percentage value of each component of the vector y. Resulting values are employed as elements of the simplex in addition to initial value i. As a result new points of the simplex are generated until the simplex diameter reaches a specified threshold.

6.2.2 Optimization Strategies

Two strategies are employed to optimize OPMM parameters with the TR and NM algorithms.

Strategy 1: the OPMM parameters are optimized sequentially. An already optimized parameter remains in the parameter vector. The subsequent parameters are optimized based on the newly optimized value of the previous parameter. For example, the value of the K_{lt} (after optimization) is employed for subsequent optimization of K_{se}, and the values of both K_{lt}, K_{se} (after optimization) are employed for subsequent optimization B_p, etc.

Strategy 2: the OPMM parameters are optimized sequentially. An already optimized parameter is saved in the temporal parameter vector and the value of this parameter in the original vector is restored to the original value. The subsequent parameters are optimized based on the original values of the remaining parameters. For example, the value of the K_{lt} (after optimization) is stored in a temporal vector and the estimation of the K_{se} occurs with the initial value of the K_{lt}. The K_{se} is stored in the temporal vector. The optimization of the B_p is based on the initial values of K_{lt}, K_{se}, etc. When all OPMM parameters are estimated the temporal vector holds the data for person identification.

7 Person Identification

The input to the Person Identification module consists of a set of OPMM parameter vectors estimated for each qualifying saccade. The output is an authorization score classifying each saccade as belonging to an authorized user or an imposter. In order to perform the classification, we evaluated two different statistical algorithms, the K-nearest neighbor (KNN) algorithm, and C4.5. KNN is a very simple instance-based learning algorithm, and C4.5 is a freely-available classifier that builds a decision tree based on the concept of information entropy [22].

The eye movement record for an individual consists of multiple saccades, and as a result the biometric identification record for each individual will consist of a set of OPMM parameter vectors. We work with the complete set of per-saccade parameter vectors, and split them into a training and a testing set to perform identification. Figure presents an example.

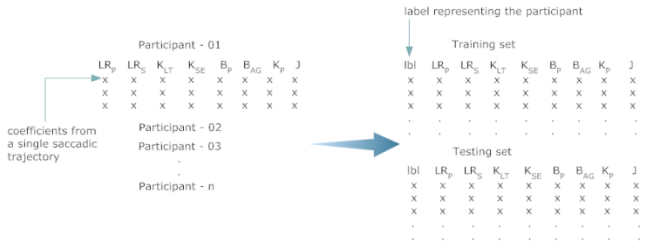

Figure 3. OPMM parameter vectors assignment to testing and training sets.

The following methodology is used to partition each participant's data into training and testing sets. Each participant data set containing exactly two records is arbitrarily declared to be an imposter, and included only with the testing set. For each participant data set containing three records or more, the parameter vectors are inserted into both testing and training sets as authorized users; the first record is inserted into the testing set and subsequent samples are inserted into the training set. We decided to split the sets in this way because of the relatively small amount of test data; however, it is possible that we have thereby introduced a systematic bias into the training set, and we plan further experiments to eliminate this possibility. We also note that samples identified as imposters, although not included in the training set, are still tracked in order to obtain the correct false acceptance rate (FAR) value for available imposters.

7.1 KNN Classification Algorithm

The *k-nearest neighbor* (KNN) algorithm [22] (in our implementation, k=5) is one of the simplest classification algorithms, but is accurate and powerful when samples with similar classification tend to appear nearby. 'Nearby' means that the distance between similar classifications is generally closer than

```
Algorithm oculomotor_KNN(training, testing, k)
    Input: Training set, Testing Set and k, where k=1,3,5,7…….
    Output: Testing set with the identified training record label
    for i←1 to size(testing) do
        for j←1 to size(training) do
            LR_P ←sqrt ((testing.LR_P(i) – training.LR_P(j) )^2)
            LR_S ←sqrt ((testing.LR_S(i) – training.LR_S(j) )^2)
            K_LT ←sqrt ((testing.K_LT(i) – training.K_LT(j) )^2)
            K_SE ←sqrt ((testing.K_SE(i) – training.K_SE(j) )^2)
            B_P ←sqrt ((testing.B_P(i) – training.B_P(j) )^2)
            B_AG ←sqrt ((testing.B_AG(i) – training.B_AG(j) )^2)
            K_P ←sqrt ((testing.K_P(i) – training.K_P(j) )^2)
            J ←sqrt ((testing.J(i) – training.J(j) )^2)
            distance(j,1) ← LR_P;        distance(j,2) ← LR_S;
            distance(j,3) ← K_LT;        distance(j,4) ← K_SE;
            distance(j,5 ← B_P;          distance(j,6) ← B_AG;
            distance(j,7) ← K_P;         distance(j,8) ← J;
            distance(j,9) ← training.lbl(j);
            [rows,columns] ← size(distance);
        for a←1 to (columns-1)
            coefficient_lbl(a) ←get_k_lbl(sorcolumns(distance,a),k)
        testing.lbl(i)←mode(coefficient_lbl);

return testing

Algorithm get_k_lbl(distance,k)
    Input: distance vector with sorted columns by each coefficient
    Output: coefficient label through k
    for i←1 to k
        distance_lbl(i) ← distance(i,columns)
    coefficient_lbl ← mode(distance_lbl);
return coefficient_lbl
```

Figure 4. Pseudocode for modified KNN algorithm

the distance between samples with different classification [4]. The pseudocode segment depicted in Figure 4 implements the oculomotor k-nearest neighbor algorithm and returns the testing set with an appropriate label assigned to each coefficient vector. For each oculomotor parameter coefficient, a distance value is obtained and recorded into a distance vector. The distance vector is processed to obtain the label value by sorting the distance vector by each corresponding oculomotor coefficient parameter and obtaining the mode value of the label among the 5 (if k=5) least distant coefficients in the vector. The returned label value from get_k_lbl() (the corresponding mode label from each oculomotor parameter coefficient based on k value) is recorded to obtain the final label value (by computing the mode again of all the labels corresponding to each oculomotor parameter coefficient) which is assigned to the testing sample after the identification process.

7.2 C4.5 Classification Algorithm

C4.5 is a classification algorithm which builds a decision tree from a training set of data, where the split at each node maximizes the information gain which represents the difference in entropy in the set after and before the split. We selected a decision tree classifier because they are robust to noisy data and C4.5 because it is widely used and easy to implement.

The input data set is a list of entries where each entry consists of values for the attributes which have been predefined for the particular data set. The split on each level is performed based on the values of a given attribute.

```
Algorithm oculomotor_c45(training)
  Input: Training set
  Output: Decision tree
  {training set empty: make a rejection leaf node}
  if size(training) = 0 then
    node.attribute ← 0
    node.value ← -1
  {one single subject}
  if number_of_subjects(training) ==1 then
    node.attribute ← 1
    node.value ← subject
  {computes the main entropy of the training set as the sum of pi *
  log(pi) where pi is the probability of finding the subject i}
  main_entropy ← compute_entropy(training);
  best_gain ← main_entropy
  {finds the attribute giving the best entropy gain after a split}
  for a ← 1 to size(attributes)
      subjects = training.get_subjects();
      {average of the value of the current attribute for each subject}
      for i ← 1 to size(subjects)
          averages(i) = subjects(i).get_average(attributes(a));
      {splits the training set: every subset contains the entries whose
      distances from a given average are the nearest}
      for i ← 1 to size(training)
          attribute_value ← training{attributes(a)}(i);
          best_distance ← Infinity;
          for j ← 1 to size(averages)
              if |attribute_value – averages(j)| < best_distance then
                  subsets(j).add(training(i));
                  best_distance ← |attribute_value – averages(j)|;
      new_entropy ← 0;
      {computes the entropy in every subset as as the sum of pi *
      log(pi) where pi is the probability of finding the subject i in the
      subset}
      for i ← 1 to size(subsets)
          new_entropy ← new_entropy +
          compute_entropy(subsets(i));
      if best_gain > (main_entropy - new_entropy) then
          best_attribute ← attribute
          best_gain ← main_entropy - new_entropy
          best_subsets ← subsets
          best_averages ← averages
  node.attribute ← best_attribute
  node.averages ← averages
  {the C4.5 algorithm is called recursively to classify the entries of
  each subset}
    for i ← 1 to size(subsets)
        node.children.add(oculomotor_c45(subsets(i)))
  return node
```

Figure 5. Pseudocode for modified C4.5 algorithm

In this work, the C4.5 algorithm was first used to build a decision tree from the training sample. To construct each node of the decision tree we first need to find the split which provides the best entropy gain. Every split is made based on one of the attributes of the data set. In order to find the best attribute, the entries from the training set are classified into several subsets. Before the classification is performed, the averages of the values of the attribute are computed for each subject. Then, each entry is added to the subset corresponding to the nearest average. The distance from the average is defined as the difference between the value of the attribute in the entry and the value of the average. The

```
Algorithm browse_tree(node, test_entry)
  Input: Root node of decision tree, the entry to identify
  Output: The subject corresponding to the entry
  {the test_entry doesn't match any classification}
  if node.attribute = 0 then
    return -1;
  {the test_entry matches the current node}
  if node.attribute = 1 then
    return node.value;
  {finds the nearest average to the current test_entry}
  attribute_value ← test_entry{node.attribute}(1);
  for i ← 1 to size(node.children)
    if |attribute_value - node.averages(i)| < nearest_distance then
      nearest_average_subnode ← node.averages(i);
      nearest_distance ← |attribute_value – node.averages(i)|;
  {browses the remaining nodes of the subtree}
  subject = browse_tree(nearest_average_subnode, test_entry);
  return subject
```

Figure 6. Subject identification by C4.5 tree browsing.

split which provides the best entropy gain is chosen. The same operation is performed on every subset of the split to form the children of the node. The process terminates when an empty subset or a subset with one single subject is found.

The next step of the algorithm is to browse the tree to identify the subject corresponding to the entry being tested. At each node, the average with the closest value to the value of the node's attribute in the entry being processed is chosen. The same algorithm is then applied to the child that corresponds to the chosen average. The process stops once a leaf node is reached. Given the value at the leaf node, a conclusion as to the person's identity can then be drawn from the value at this node. Figure 5 provides the pseudocode.

Once the decision tree is built, it is browsed in order to identify the subject in each of the entries from the testing set. The pseudocode presented by Figure 6 is employed to browse the decision tree.

8 EXPERIMENTAL METHODOLOGY

8.1 Apparatus

The experiments were conducted with a Tobii x120 eye tracker [23], which consists of a standalone unit connected to a 24-inch flat panel screen with resolution of 1980x1200. The eye tracker performs binocular tracking with the following characteristics: accuracy 0.5°, spatial resolution 0.2°, drift 0.3° with eye position sampling frequency of 120Hz. The Tobii x120 model allows 300x220x300 mm freedom of head movement. Nevertheless, a chin rest was employed in our experiments to provide higher accuracy and stability. Subjects were seated approximately 710 mm from the eye tracker interface.

8.2 Procedure

8.2.1 Accuracy test

An accuracy test was employed prior to the experiment providing us with average calibration error and invalid data percentage for each subject. The accuracy test is described in more detail in [24].

8.2.2 Eye Movement Invocation Task

The stimulus was presented as a 'jumping point' with vertical coordinate fixed to the middle of the screen. The first point was presented at the middle of the screen; the subsequent points moved horizontally to the left and to the right of the center of the screen with a spatial amplitude of 20°, providing average stimuli amplitude of approximately 19.3°. The jumping sequence consisted of 15 points including the original point in the center, yielding 14 saccades for each participant. After each jump, the point remained stationary for 1.5s before the next jump was initiated. The size of the point was approximately 1° of the visual angle with the center marked as a black dot. Each point consisted of white pixels (except for the central black dot), with the remainder of the screen left black.

8.3 Participants

The test data consisted of 68 student volunteers ages 18-25 with an average age of 21.22 and standard deviation of 3.23, 24 males and 44 females, with normal or corrected-to-normal vision. None of the participants had prior experience with eye tracking. The data collection was verified to be accurate by employing two parameters, the average calibration error of the right eye and the invalid data percentage of the right eye. The data analyzer was instructed to discard recordings from subjects with a calibration error of $> 3.0^{\circ}$, with mean of 1.25°, standard deviation of 0.77° and invalid data percentage of >50%. Only 41 subject records passed these criteria, resulting in mean accuracy of 1.25º (SD=0.77°) and a mean invalid data percentage of 12.43% (SD=17.22%). Only saccades with amplitudes of 17-22º were employed for biometric identification.

8.4 Performance evaluation metrics

Performance evaluation of a biometric system is measured with the following two parameters.

False Acceptance Rate ***(FAR)*** – The ratio of the number of imposter samples classified as authentic to the total number of all the imposter samples. This metric measures the probability that the system incorrectly matches the input pattern of the testing set to a non-matching template in the training set. It measures the percent of invalid inputs which are incorrectly accepted.

False Rejection Rate ***(FRR)*** – The ratio of the number of authentic samples classified as imposters to the number of all the authentic samples. This metric calculates the probability that the system fails to detect a match between the input pattern of the testing set and a matching template in the training set. It measures the percent of valid inputs which are incorrectly rejected.

9 Results

We conducted the classification with both the KNN and C4.5 algorithms on each of the OPMM parameters, and determined that the best results were obtained with KNN utilizing the TR algorithm with optimization strategy 1 for the length tension coefficient. These results improve on previous work in the field by Kasprowski (Kasprowski 2008). C4.5 did not produce acceptable results with the tested algorithm parameters. We present tables of all our results below.

9.1 KNN

Figure 7 presents the KNN results.

Optimization	Metric	LR_P	LR_S	K_{LT}	K_{SE}	B_P	B_{AG}	K_P	J	D
TR-S1	FAR	26.3%	26.3%	**5.3%**	26.3%	21.1%	26.3%	36.8%	42.1%	26.3%
TR-S1	FRR	66.6%	64.5%	**56.6%**	66.6%	66.6%	70.0%	73.3%	76.6%	70%
TR-S2	FAR	31.6%	36.8%	21.1%	26.3%	36.9%	31.6%	31.6%	21.1%	26.6%
TR-S2	FRR	73.3%	76.7%	70%	63.3%	80%	73.3%	76.7%	70.0%	68.9%
NM-S1	FAR	26.3%	36.8%	26.3%	42.1%	26.3%	26.3%	**15.8%**	47.4%	**15.9%**
NM-S1	FRR	68.8%	78.1%	71.9%	81.3%	71.9%	68.8%	**62.5%**	81.3%	**62.5%**
NM-S2	FAR	31.6%	31.6%	26.3%	42.1%	26.3%	21.1%	36.8%	31.6%	36.9%
NM-S2	FRR	71.9%	71.9%	68.8%	81.3%	68.8%	68.8%	75%	75%	78.1%

Figure 7. Biometric identification via OPMM with KNN (values in bold are the three best obtained in this experiment)

The optimization column contains the abbreviated name for the optimization algorithm and optimization strategies, i.e., TR stands for Trust Region and NM stands for Nelder-Mead, S1 for optimization strategy 1 and S2 for optimization strategy 2 as described in section 6.2.2. The header row lists the elements of the oculomotor plant parameter vector, where the last value "D" is the Euclidean distance computed between all components of the oculomotor plant parameter vector.

The first three best results are in bold. The smallest FAR and FRR values were achieved with the Trust-Region algorithm using optimization strategy 1 for the length tension coefficient (K_{LT}). The two next best results were provided by the Nelder-Mead algorithm using optimization strategy 1 for the passive elasticity coefficient (K_p) and the distance created by all parameters (D). The FAR=5.4% and FRR=56.6% results improve on the previously reported results of FAR=9.4% and FRR=63.4% given by Kasprowski in 2004 using the KNN algorithm [4].

9.2 C4.5

Optimization	Metric	C4.5
TR-S1	FAR	87.2%
TR-S1	FRR	4.25%
TR-S2	FAR	79.5%
TR-S2	FRR	4.1%
NM-S1	FAR	94.1%
NM-S1	FRR	2%
NM-S2	FAR	**80%**
NM-S2	FRR	**0%**

Figure 8. Biometric identification via OPMM with C4.5

The first three best results are in bold. Unfortunately, we did not obtain good FAR rates with the C4.5 algorithm, even if we accepted an increase in FRR. We believe the FAR can be improved by further tuning of the algorithm parameters. The best FAR=80% and FRR=0% values were achieved by Nelder-Mead with optimization strategy 2. For comparison, Kasprowski obtained an FAR of 45.8% and an FRR of 12.4% using C4.5 in 2004 [4].

10 Discussion, Conclusions and Further Work

We have introduced a novel method of biometric identification based on the utilization of Oculomotor Plant Mathematical Model parameters from horizontal positive saccadic eye movements. We evaluated the effectiveness of this method via two different statistical classification techniques on a data set of horizontal saccadic eye trajectories collected from 41 human participants, and achieved promising results using the k-nearest neighbor classification algorithm. Our results improve on previous biometric methods involving eye movements.

The OPMM method of biometric identification leverages physiological and behavioral characteristics that are unique to each individual – the mechanical properties of the eye globe and its musculature – rather than simply looking at unprocessed saccadic trajectories. The resulting additional information provides further structure to the eye movement data and perhaps this is what leads to the improved performance of our method over previous work. A further advantage of the proposed method is its use of a dynamic oculomotor plant model consisting of the eye globe and extraocular muscles that is extremely difficult to counterfeit.

Via our tests, we demonstrated the potential to distinguish authorized users from imposters with this technique. However, further testing with larger subject pools and different statistical classification algorithms is needed to improve on the accuracy rates of our method. Nevertheless, this technique shows promise for improving the state of biometric identification. This new method could also be easily combined with existing biometric identification systems that incorporate digital cameras to scan the face or iris, to provide an additional layer of security. In an ever-more security-conscious and highly networked world, non-intrusive and unforgeable personal identity-based authorization methods will become increasingly critical across a wide range of commercial and government applications.

11 References

[1] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies," *ACM Communications,* vol. 33, pp. 168--176, 1990.

[2] S. Wiedenbecka, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "PassPoints: Design and longitudinal evaluation of a graphical password system," *International Journal of Human-Computer Studies,* vol. 63, pp. 102-127, July 2005.

[3] B. Schneier, "Two-factor authentication: too little, too late," *ACM Communications,* vol. 48, p. 136, 2005.

[4] P. Kasprowski, "Human identification using eye movements," in *Faculty of Automatic Control, Electronics and Computer Science*. vol. Ph.D. Gliwice: Silesian University of Technology, 2004, p. 111.

[5] J. Daugman, "How iris recognition works," in *Image Processing. 2002. Proceedings. 2002 International Conference*, 2002, pp. I-33- I-36.

[6] A. Jain, L. Hong, and Y. Kulkarni, "A Multimodal Biometric System Using Fingerprint, Face, and Speech," in *In International Conference on AVBPA*, 1999, pp. 182-187.

[7] V. Chatzis, A. G. Bors, and I. Pitas, "Multimodal decision-level fusion for person authentication," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol. 29, pp. 674-680, 1999.

[8] S. Josephson and M. E. Holmes, "Visual attention to repeated internet images: testing the scanpath theory on the world wide web," in *In ETRA 02: Proceedings of the 2002 symposium on Eye tracking research & applications*, 2002, pp. 43-49.

[9] J. M. Williams, "Biometrics or ... biohazards?," in *Proceedings of the 2002 workshop on New security paradigms*, Virginia Beach, Virginia, 2002, pp. 97-107.

[10] L. Wiskott, M. Fellous, and N. Kruger, "Face recognition by elastic bunch graph matching," in *International Conference on Image Processing (ICIP 97)* Los Alamitos, CA, USA, 1997.

[11] W. Zhao, R. Chellappa, and P. J. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys (CSUR),* vol. 35, pp. 399-458, 2003.

[12] O. Komogortsev, V. and J. Khan, "Eye Movement Prediction by Kalman Filter with Integrated Linear Horizontal Oculomotor Plant Mechanical Model," in *Eye Tracking Research & Applications Symposium 2008*, Savannah, GA, 2008, pp. 229-236.

[13] A. T. Bahill, "Development, validation and sensitivity analyses of human eye movement models," *CRC Critical Reviews in Bioengineering,* vol. 4, pp. 311-355, 1980.

[14] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye tracking protocols," in *Eye Tracking Research and Applications Symposium*, New York, 2000, pp. 71-78.

[15] D. L. Sparks, "The brainstem control of saccadic eye movements," *Nat. Rev. Neurosci.,* vol. 3(12), pp. 952-964, 2002.

[16] O. Komogortsev and J. Khan, "Eye movement prediction by oculomotor plant Kalman filter with brainstem control," *Journal of Control Theory and Applications,* vol. 7, pp. 14-22, 2009.

[17] D. R. Wilkie, "Muscle," *Studies in Biology,* vol. 11, 1976.

[18] C. C. Collins, "The human oculomotor control system," *Basic Mechanisms of Ocular Motility and Their Applications,* pp. 145-180, 1975.

[19] D. A. Robinson, D. M. Omeara, A. B. Scott, and C. C. Collins, "Mechanical components of human eye movements," *Journal of Applied Physiology,* pp. 548-553, 1969.

[20] T. F. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM Journal on Optimization,* vol. 6, pp. pp. 418–445, 1996.

[21] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder--Mead Simplex Method in Low Dimensions," *SIAM J. on Optimization,* vol. 9, pp. 112-147, 1998.

[22] G. Shakhnarovich, T. Darrell, and P. Indyk, "Nearest-Neighbor Methods in Learning and Vision," in *Neural Information Processing*: MIT Press, 2005.

[23] Tobii, "Tobii technology," 2009.

[24] D. H. Koh, S. A. M. Gowda, and O. V. Komogortsev, "Input evaluation of an eye-gaze-guided interface: kalman filter vs. velocity threshold eye movement identification," in *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems* Pittsburgh, PA, USA: ACM, 2009, pp. 197-202.