IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 7, 2021, accepted March 26, 2021, date of publication April 1, 2021, date of current version April 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3070511

Saccade Landing Point Prediction Based on Fine-Grained Learning Method

AYTHAMI MORALES[1,2], FRANCISCO M. COSTELA[2,3], AND RUSSELL L. WOODS[2,3]

[1]BiDA-Lab, Department of Electrical Engineering, Universidad Autonoma de Madrid, 28049 Madrid, Spain
[2]Schepens Eye Research Institute, Massachusetts Eye and Ear, Boston, MA 02114, USA
[3]Department of Ophthalmology, Harvard Medical School, Boston, MA 02115, USA

Corresponding authors: Aythami Morales (aythami.morales@uam.es), Francisco M. Costela (francisco_costela@meei.harvard.edu), and Russell L. Woods (russell_woods@meei.harvard.edu).

This work was supported in part by the Project BIBECA through MINECO/FEDER under Grant RTI2018-101248-B-100, in part by the Jose Castillejo Program through MINECO under Grant CAS17/00117, and in part by the National Institutes of Health (NIH) under Grant P30EY003790 and Grant R21EY023724.

ABSTRACT The landing point of a saccade defines the new fixation region, the new region of interest. We asked whether it was possible to predict the saccade landing point early in this very fast eye movement. This work proposes a new algorithm based on LSTM networks and a fine-grained loss function for saccade landing point prediction in real-world scenarios. Predicting the landing point is a critical milestone toward reducing the problems caused by display-update latency in gaze-contingent systems that make real-time changes in the display based on eye tracking. Saccadic eye movements are some of the fastest human neuro-motor activities with angular velocities of up to 1,000°/s. We present a comprehensive analysis of the performance of our method using a database with almost 220,000 saccades from 75 participants captured during natural viewing of videos. We include a comparison with state-of-the-art saccade landing point prediction algorithms. The results obtained using our proposed method outperformed existing approaches with improvements of up to 50% error reduction. Finally, we analyzed some factors that affected prediction errors including duration, length, age, and user intrinsic characteristics.

INDEX TERMS Saccade, eye movement, gaze-contingent, recurrent neural networks, LSTM, fine-grained learning.

I. INTRODUCTION

Gaze-contingent displays [1] have been used in video streaming [2], [3], robot-assisted surgery [4], human-computer interfaces for new virtual reality environments [5], reading research [6] and simulation of impaired vision [7], [8], among others. These systems allow researchers to investigate a variety of visual phenomena, including eye movement guidance in reading, stability of vision, visual search strategies, and scene perception. Due to data transmission, image processing, and data display preparation, all systems have an updating latency of at least 10 ms [9] to 12 ms [10] that introduces an updating error (see Figure 1).

Saccades are very fast eye movements made between phases of fixation which bring the image of the region of interest to the fovea, the small (1 to 2°) retinal region with the highest resolution. During typical viewing, we make about three saccades per second. Saccades are one of the fastest muscle movement, up to 1,000°/s (peak angular velocity). Saccades have short durations, lasting about 15 to 200ms, with the duration closely tied to the length (amplitude) and to the peak velocity. While the common perception is that saccades are ballistic eye movements that are not modified during flight and have a straight path, this is often not the case [11]. Since saccades are fast, due to the unavoidable latency of gaze-contingent systems (that are intended to update displays according to gaze location), the gaze is no longer at the measured location by the time the display can be updated.

Saccades are responses of the visual system mainly to visual stimuli. One approach to predicting the next saccade landing point is to analyze the visual scene for the most likely new region of interest. These visual salience models have improved over recent decades [12]. The computations for visual salience may be challenging for a real-time application and a scene that is in motion. Another limitation of the visual

The associate editor coordinating the review of this manuscript and approving it for publication was Giuseppe Desolda.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

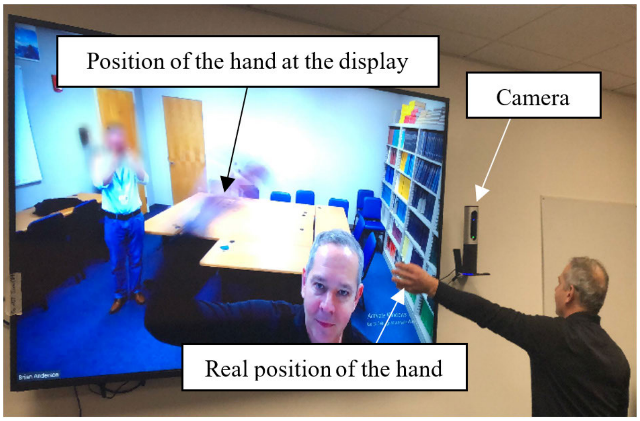
Position of the hand at the display
Camera
Real position of the hand

FIGURE 1. An illustration of system latency in which the virtual arm (on display) is shown in an earlier location of the real arm as the arm was extended. The large television (Vizio M80-D3) was running at 60Hz frame rate. The system latency was about 30ms, which is low for a digital display [10].

salience approach is that such models have not been shown to be able to predict the timing of saccade initiation, so could not inform the timing of the change of a gaze-contingent display. As an alternative approach, we propose using the early part of a tracked saccade to predict the landing point. Such an approach is agnostic to visual stimuli, employing saccade dynamics alone.

Modelling saccade dynamics is challenging, as saccades may not be ballistic or straight [11]. The relationship between saccade amplitude and maximum velocity (the saccadic main sequence) was first described by Bahill *et al.* [13]. Longer saccades have been modeled with lognormal functions, while Gaussian functions [14], or compressed exponential functions [15] may be better for shorter saccades. Various models have been proposed or used to predict saccade amplitude, direction or trajectory, including mirroring the data points before the peak velocity [14], the main sequence [16], [17], Kalman filters [18]–[20], a compressed exponential function [15], a Taylor series [21], and a skewed Gaussian function [22]. Saccade prediction could improve the experience with gaze-contingent displays [15], [18]–[21].

Each saccade landing point determines the new fixation or gaze location, with that object being imaged at the fovea. Prediction of landing point may reduce problems due to the system-update latency of gaze-contingent displays, as the display could be updated at the predicted saccade landing point while the saccade was in flight [22]–[26]. Due to the speed of saccades, vision is very poor during the saccade [27], so an update is unlikely to be noticed. Previous approaches to predicting saccade landing location include polynomial fitting [24], Recurrent Neural Networks (RNNs) [25], and velocity profile fitting [22]. Here we present a new method for landing-point prediction.

Figure 2 shows an example of a measured saccade path captured during natural viewing of a video clip. The figure shows the path when the region of interest changed from stimulus A (the man) to stimulus B (the woman). In saccade landing point prediction based on eye tracker data, the prediction is performed every time a saccade starts. The goal is to predict the target gaze position (defined by the landing point of the saccade) using the first coordinates provided by the eye tracker. The prediction is performed using exclusively the eye tracker data (i.e. the stimuli are unknown). The landing point prediction can be updated with every new position provided by the eye tracker. The figure illustrates some of the main challenges associated with saccade-landing-point prediction including: i) nonlinear trajectory of the movement; ii) measurement noise of the eye tracker; and iii) nonlinear velocity profiles (e.g. the distance between data samples during the first 10ms and the last 38ms varies). Figure 3 illustrates the target gaze position (horizontal axis only), the delay introduced by the hardware, and the prediction of the landing point (horizontal coordinate) obtained with the proposed prediction updated every 10 ms.

Among the disruptive technologies of last years, Deep Learning (DL) has become a thriving topic [28]. The increasing volumes of available data and the increased computational power are used to model complex problems in terms of a hierarchy of simpler processing units. Data-driven learning methods has allowed advances in many fields such as Computer Vision [29], Speech Recognition [30], and Natural Language Processing [31].

Recurrent Neural Networks (RNN) are architectures especially useful when modeling sequences [32]. The number of applications of RNNs is large and it includes modeling sequences of data with very different natures such as handwritten-signature recognition [33], [34], audio processing [30], [35], and epilepsy [36]. RNN architectures incorporate dynamic temporal relationships by self-connected units (i.e. recurrent connectivity). This connectivity serves to incorporate temporal context into the learning process. The context can be passed through units or stored in specific memory units as an internal state. However, traditional RNN are not capable of maintaining this contextual information for long term temporal relationships. This problem is known as vanishing gradient problem [37]. Long Short-Term Memory (LSTM) [38] was proposed trying to solve the vanishing problem. LSTM networks incorporate specific memory units to model long-term temporal relationships.

A loss function is used to guide the learning process of data-driven approaches. Choosing the loss function that best fits the modelling task at hand is critical. The number of loss functions in the literature is large. Among the different learning strategies, fine-grained learning is aimed to distinguishing sub-categories inside a primary category. Fine-grained learning merges sub-ordinate patterns allowing improved classification [39] and regression models [40].

The main contributions of our work are:

- A new saccade landing point prediction method based on LSTMs and a fine-grained learning strategy.
- A comprehensive analysis of the performance over more than 200,000 saccades from 75 participants acquired during viewing video. This analysis includes the effects

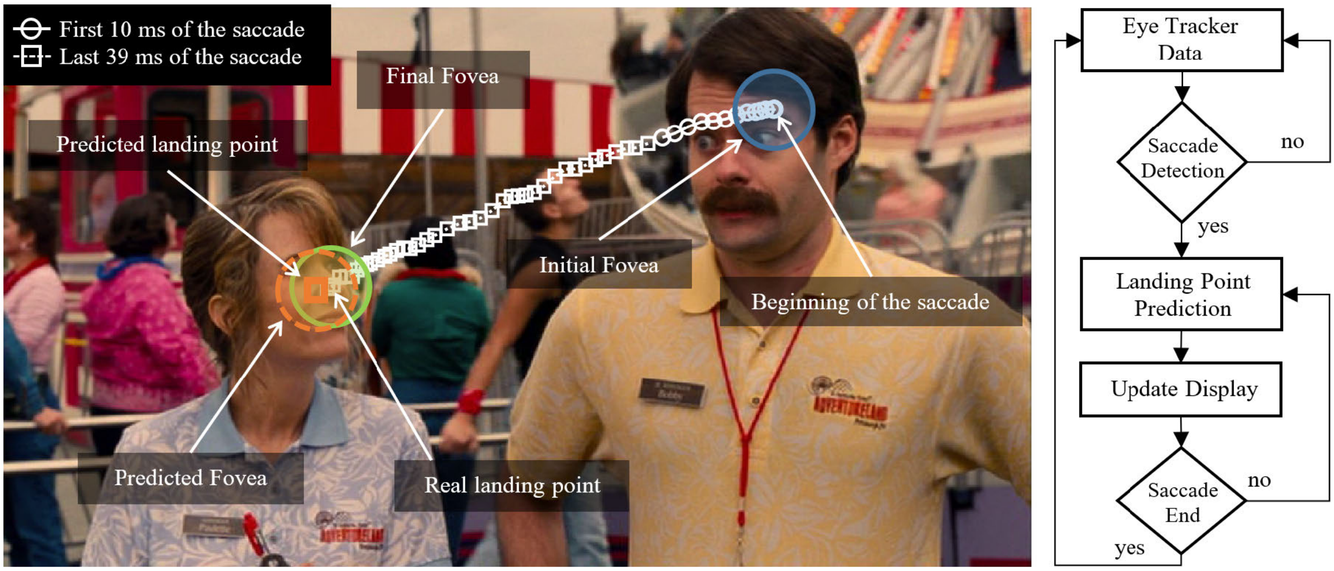

FIGURE 2. Landing point prediction from a 10 ms sequence (small white circles) from a real saccade movement acquired with an EyeLink 1000 eye tracker (1,000Hz). The trajectory of the 49ms saccade is represented by the Cartesian coordinates of each of the samples obtained from the eye tracker (small white circles and squares). The gaze positions are in the region of interest of the viewer, with the fovea represented by the colored circles, each with a diameter of 2.5°. Note that prediction is performed independently of the stimuli (i.e. video frames are not used in the prediction).

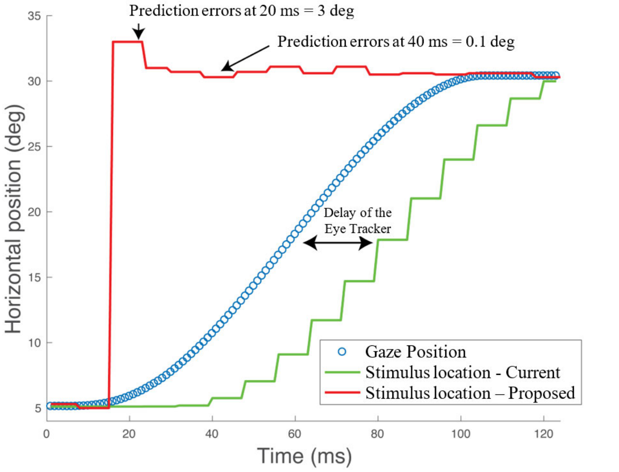

FIGURE 3. Gaze position (blue) during one left-to-right saccade. Updating of a gaze-contingent stimulus is shown for simple updating which was delayed by the system latency (green line) on an iMac Intel Core i7 (60 Hz display) connected to an Eyelink 1000. The red line used our proposed method that predicted the landing point of the saccade and moved the stimulus to the predicted landing location so that it was visible immediately on saccade landing. The approach assumes that the stimulus is not visible during a saccade ("saccadic suppression").

of covariates including user-dependent (e.g. neuromotor physiognomy) and task-dependent (e.g. number of samples available to make the prediction, length of the saccade) characteristics.

- Comparison with state-of-the-art approaches [24], [25]. The experiments demonstrate the superior performance of the fine-grained learning approach in comparison with traditional learning functions. The results obtained using our proposed method outperformed existing approaches with improvements ranging from 20% to 50% error reduction.

The rest of this paper is organized as follows: the end of Section 1 presents the related works. Section 2 describes the saccade landing point prediction approach, the learning strategy, and the database used for the experimental protocol. Section 3 reports the experiments and results. Finally, Section 4 summarizes the conclusions.

A. RELATED WORKS

There have been previous efforts on predicting the saccade landing point from eye tracker data [22], [24]–[26]. The method used in [24] was based on polynomial fitting trained to predict displacement and angle of the saccade. The results reported in [24] suggested that it is possible to predict the landing point of saccades and its potential to improve gaze-rendering systems. The method in [22] was based on velocity profile prediction, using the close relationship between velocity and amplitude [13]. They proposed fitting a Gaussian velocity function to predict the amplitude of the saccade. Both studies [24], [22] used eye-movement data obtained from non-natural viewing conditions, with saccades made to pre-defined visual stimuli. The prediction of saccade landing point using RNNs was proposed in [25], [26] where researchers evaluated Long Short-Term Memory Networks (LSTM). The results in [25] demonstrated the capacity of these networks to model the saccade movement and outperform function-fitting approaches in natural viewing. These approaches try to exploit the predictable characteristics of the oculomotor system. The aim is modeling the movement using only the early part of the saccade.

As noted above, a potential alternative approach uses visual attention, in which the goal is to predict group behavior or typical behavior of people within a group [12]. These methods are usually aimed to model mid-term user behavior rather than individual saccade movements. Visual attention models can be divided into two categories: *i*) Static saliency maps prediction based on static fixations [41]–[44]; and *ii)* Dynamic saliency maps prediction incorporating the temporal sequence of fixations [45]–[47]. It is not clear that visual attention models will be able to provide the timing of saccade events at the individual level or to handle situations when there are two equally salient new locations (as in Figure 2) that produce bimodal gaze distributions (e.g. Figure 1 in [48]) as required for gaze-contingent and virtual-reality systems.

II. MODELING SACCADIC EYE MOVEMENTS

A. NATURAL VIEWING DATABASE

Data was drawn from a shared database [49]. It comprised the gaze of 75 human participants (37 female, 38 male) with normal sight who participated in studies approved by the Institutional Review Board of the Schepens Eye Research Institute and were in accordance with the Declaration of Helsinki (Code of Ethics of the World Medical Association). Participant screening included self-reported ocular health, measures of central visual function (visual acuity and contrast sensitivity), and evaluation of central retinal health and fixation stability (retinal imaging using a Nidek MP-1, Nidek Technologies, Vigonza, Italy or an Optos OCT/SLO, Marlborough, MA, USA). Participants had visual acuity of 6/7.5 (0.10 logMAR) or better, letter contrast sensitivity of 1.67 log units or better, no evidence of retinal defects and steady central fixation.

Participants either watched:

1. 40 to 46 of 206 thirty-second "Hollywood" video clips. Those included a variety of genres and depicted activities. The genres included nature documentaries (e.g., *Food, Inc*, *The March of the Penguins*), cartoons (e.g., *Cloudy with a Chance of Meatballs, The Simpsons Movie*), and dramas (e.g., *Stardust, The Stepfather*). Participants were instructed to watch each video clip "normally, as you would watch television or a movie program at home" and then to describe the contents of the clip [50], [51]. This group of 62 participants contributed 108,640 saccades.
2. two to five 30-minute movie clips (*Bambi, Flash of Genius, Inside Job, Juno,* and *Kpax*). This group of 14 participants (one was also in the first group) contributed 110,695 saccades.

A detailed description and additional information about the database can be found in [49].

Figure 4 presents the database statistics for duration and length of the saccades, and age of the participants. As we will see in the experimental section, these factors are important to determine the prediction error. The average duration was 34 milliseconds, while average length was 5.8 degrees. The average age of the participants in the database was 54 years old with similar number of participants younger and older than 60 years old.

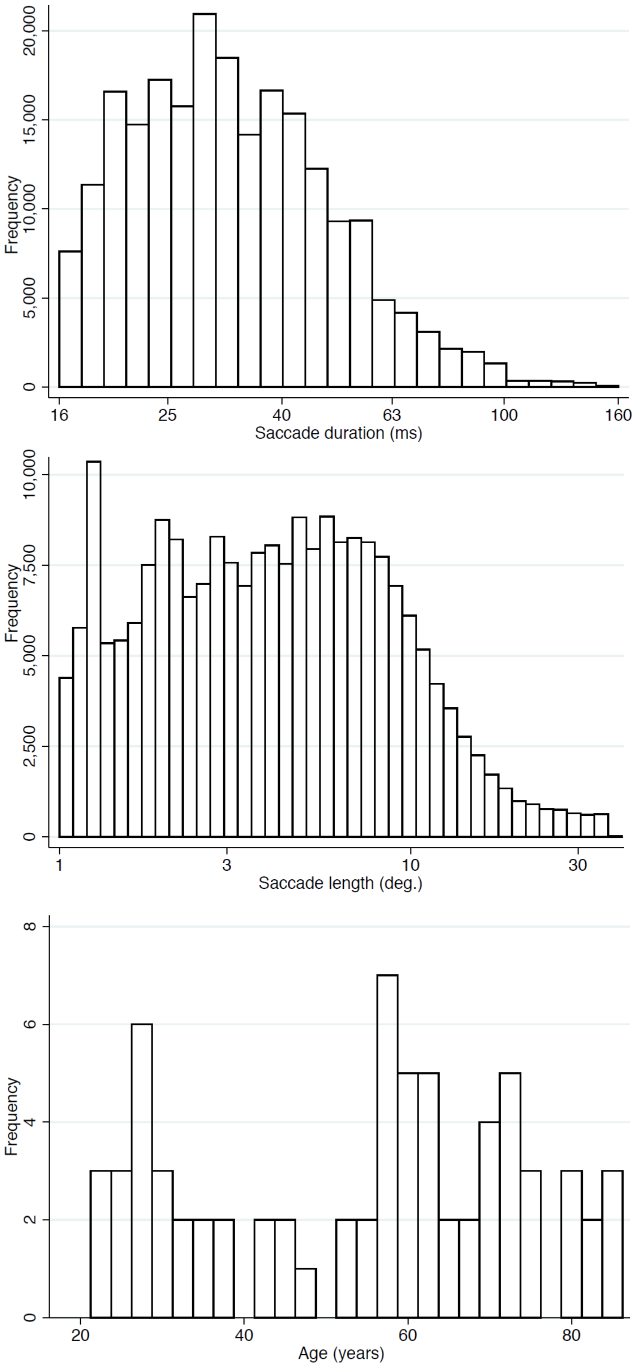

FIGURE 4. Descriptive histograms showing the duration and length (amplitude) of the 219,335 saccades, and the ages of the 75 participants.

B. SACCADE DETECTION

Gaze was tracked with an EyeLink 1000 infra-red video-based eye tracker (SR Research Ltd., Mississauga, ON, Canada) at a sampling rate of 1,000 Hz while participants viewed a 27" display (60 × 34 cm) from 1 m for a 33 × 19° potential viewing area. Using the method described in [21], 219,335 saccades were detected off-line. Raw data[1] were

[1] In this work, raw data refers to the data directly provided by the Eye Tracker.

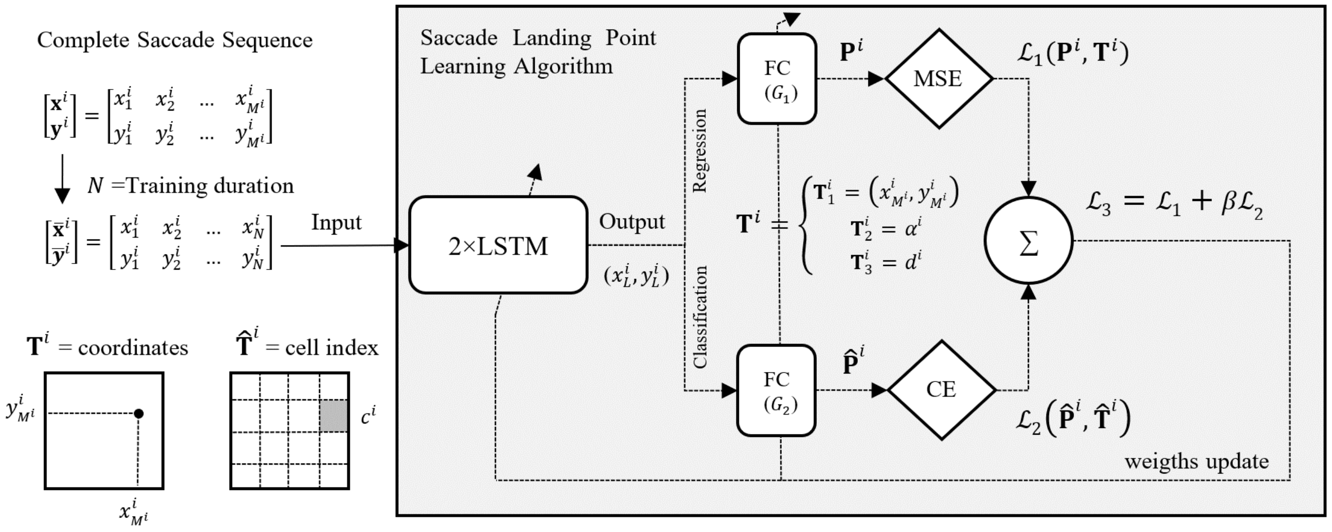
Complete Saccade Sequence
$\begin{bmatrix}\mathbf{x}^i\\ \mathbf{y}^i\end{bmatrix} = \begin{bmatrix}x_1^i & x_2^i & \dots & x_{M^i}^i\\ y_1^i & y_2^i & \dots & y_{M^i}^i\end{bmatrix}$
N =Training duration
$\begin{bmatrix}\bar{\mathbf{x}}^i\\ \bar{\mathbf{y}}^i\end{bmatrix} = \begin{bmatrix}x_1^i & x_2^i & \dots & x_N^i\\ y_1^i & y_2^i & \dots & y_N^i\end{bmatrix}$
Input
$\mathbf{T}^i$ = coordinates
$\hat{\mathbf{T}}^i$ = cell index
$y_{M^i}^i$
$x_{M^i}^i$
c^i
Saccade Landing Point Learning Algorithm
2×LSTM
Output
(x_L^i, y_L^i)
Regression
Classification
FC (G_1)
FC (G_2)
$\mathbf{P}^i$
$\hat{\mathbf{P}}^i$
MSE
CE
$\mathcal{L}_1(\mathbf{P}^i, \mathbf{T}^i)$
$\mathcal{L}_2(\hat{\mathbf{P}}^i, \hat{\mathbf{T}}^i)$
$\mathbf{T}^i = \begin{cases}\mathbf{T}_1^i = (x_{M^i}^i, y_{M^i}^i)\\ \mathbf{T}_2^i = \alpha^i\\ \mathbf{T}_3^i = d^i\end{cases}$
Σ
$\mathcal{L}_3 = \mathcal{L}_1 + \beta\mathcal{L}_2$
weigths update

FIGURE 5. Saccade landing point learning algorithm was based on three loss functions: $\mathcal{L}_1$ was obtained as the Mean Square Error (MSE) between the Prediction (P) and the Target function (T); $\mathcal{L}_2$ was obtained as the Cross Entropy (CE) between the discretized Prediction ($\hat{\mathbf{P}}$) and the discretized Target ($\hat{\mathbf{T}}$); finally, $\mathcal{L}_3$ was obtained as a linear combination of $\mathcal{L}_1$ and $\mathcal{L}_2$. FC = Fully connected layer.

used to develop and test the algorithms. EyeLink's online data parser was used to identify and remove blinks. Periods surrounding the missing data were removed if the speed was >30°/s. Missing data for blinks were replaced by interpolation using cubic splines. Before saccade detection, the raw data was smoothed with a 3rd-order Savinsky-Golay filter with a window size of 15. This smoothing made saccade detection more reliable. Speed was calculated as the first derivative of the eye position with respect to time. Saccade commencement was signaled by speed >30°/s for at least 10 ms. Saccade completion was signaled by speed <30°/s. Saccades were restricted to: (1) <40°, as this was approximately the maximum diagonal dimension of the display; and (2) >1° in amplitude and 15 ms in duration to exclude microsaccades. Additional restrictions of an initial speed <0.075°/ms, terminal speed <0.3°/ms, and removal of saccades with a velocity at first quartile of duration <0.15 peak velocity, removed eye movements with uniform but unrealistically low velocity profiles during their initial phase, which may have been pursuit eye movements. The smoothed data of the saccades that were identified using the above procedure were then replaced with the raw data. The rationale was that a real-time algorithm would have access to raw data, thus our input was realistic.

C. SACCADE LANDING POINT PREDICTION BASED ON FINE-GRAINED LEARNING STRATEGY

Saccades are complex neuromotor eye movements involving six muscles per eye, acting in a coordinated manner in a very short time. The aim of a saccade lading point prediction algorithm is to estimate the final foveated region using the first samples of the saccade obtained by the eye. The landing point is defined according to: (i) prediction model (LSTM Neural Network in our approach) and; (ii) input data provided by the eye tracker. The length, duration, and curvature of the sequence varied depending on the oculomotor setup of the movement. To model the non-monotonic trajectories of the saccade, we update the prediction every 5ms with new data provided by the eye tracker. The problem of the latency of gaze-contingent systems will be partially solved if the prediction is sufficiently accurate early in the saccade (e.g. first 15ms).

In addition to our proposed approach, we have evaluated different learning strategies to model saccades. Figure 5 shows our learning algorithm, which can be summarized in:

1) TARGET FUNCTION (T)

We evaluated two different strategies based on three Target functions ($\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$). The first strategy consisted of modelling the Cartesian coordinates of the landing point: $\mathbf{T}_1 = (x_M, y_M)$. The output of the network (x_L, y_L) was direct prediction of the position of the landing point. The second strategy included the training of a specific network for each of the Target functions ($\mathbf{T}_2$, $\mathbf{T}_3$). The first network was trained to predict the angle/direction of the saccade movement ($\mathbf{T}_2 = \alpha$), while the second network was trained to predict the displacement ($\mathbf{T}_3 = d$). The output of both networks was combined with the initial points (x_1, y_1) to obtain the landing position ($x_L = x_1 + d\cos\alpha$,$y_L = y_1 + d\sin\alpha$).

2) LOSS FUNCTIONS ($\mathcal{L}$)

Inspired by the learning strategy proposed in [40] we defined three loss functions. The first loss function ($\mathcal{L}_1$) performed a regression loss trying to obtain a fine prediction of the landing point. The LSTM blocks were connected to a dense output layer with G_1 units ($G_1 = 2$ for $\mathbf{T}_1$, $G_1 = 1$ for $\mathbf{T}_{2-3}$) and linear activation. The loss function was calculated as the l^2

norm between the Target and Predicted distributions:

$$\mathcal{L}_1(\mathbf{P}, \mathbf{T}) = \|\mathbf{T} - \mathbf{P}\| \tag{1}$$

The second loss function ($\mathcal{L}_2$) exploited the stability of the training procedure based on a softmax layer (categorical cross-entropy loss). The Target function ($\mathbf{T}$) was discretized ($\hat{\mathbf{T}}$) in G_2 classes, and the loss function was calculated as:

$$\mathcal{L}_2\left(\hat{\mathbf{P}}, \hat{\mathbf{T}}\right) = -\sum_{c=1}^{G_2} \hat{\mathbf{T}}_c \log\left(\hat{\mathbf{P}}_c\right) \tag{2}$$

where G_2 was the number of classes for each Target function ($G_2 = 64$ for landing point $\mathbf{T}_1$, $G_2 = 32$ for displacement and angle $\mathbf{T}_{2-3}$), log was the natural logarithmic, $\hat{\mathbf{T}}_c$ was a discrete label associated to class c, and $\hat{\mathbf{P}}_c$ was the predicted probability for the class c. For the discretization, the screen was divided into a 64 cells grid and each landing coordinate was replaced by the index of the nearest cell (see Figure 5). The displacement and angles were discretized into 32 equidistant values (from 0 to π for angle, from 0.1 to 40 degrees for displacement).

Finally, the third loss function ($\mathcal{L}_3$) was calculated as a linear combination of previous ones with a weighting parameter β (empirically set to 0.7):

$$\mathcal{L}_3 = \mathcal{L}_1 + \beta\mathcal{L}_2 \tag{3}$$

3) ARCHITECTURE OF THE NETWORK

Different architectures were evaluated during the experiments. The number of layers was chosen to maximize the prediction accuracy for the minimum number of parameters possible. The final neural network architecture was based on two LSTM hidden layers, and a fully connected dense layer with a linear activation ($\mathbf{T}$) and softmax activation ($\hat{\mathbf{T}}$). The number of units of the output dense layer (G_i) varies depending on the loss function $\mathcal{L}_i$ and the target function (see loss function explanation and Figure 5).

4) TRAINING PROCEDURE

As gaze was acquired at 1,000Hz, we obtained one sample every 1ms. Throughout the paper we use time units to refer to the available samples. We trained six models that varied in the time available to make the prediction. The six models were trained to predict the landing point from given sequences with only the first 10ms, 15ms, 20ms, 25ms, 30ms or 35ms. The proposed architecture was trained using the Cartesian coordinates of the training saccades, according to the following steps:

a. Each of the saccade training sequences $(\mathbf{x}^i, \mathbf{y}^i)$ with length M^i were truncated with length $N = 10, 15, 20, 25, 30, 35)$ depending on which of the six models was trained. Zero padding was applied when $M^i < N$.

b. The target function $\mathbf{T}$ was chosen depending of the output to be modeled: landing coordinates $(\mathbf{T}_1 = (x^i_M, y^i_M))$, angle $(\mathbf{T}_2 = \alpha^i)$, or displacement $(\mathbf{T}_3 = d^i)$.

c. The loss function was chosen depending of the learning strategy: mean square error between Prediction and Target ($\mathcal{L}_1$); cross entropy between discretized values of Prediction and Target ($\mathcal{L}_2$); linear combination of both losses ($\mathcal{L}_3$).

d. The architecture of the network was built with two LSTM layers (each consisting of 32, 64, or 128 units depending of the setup evaluated, see Table 1) and a dense layer (G units with linear activation for $\mathbf{T}_1$, and softmax activation for $\mathbf{T}_2$ and $\mathbf{T}_3$). Additionally, we evaluated both forward and bidirectional learning.

e. The truncated training sequences $(\bar{\mathbf{x}}^i, \bar{\mathbf{y}}^i)$ and the target output $\mathbf{T}$ were used to train each of the six models (different values of N). All LSTMs were trained using a Backpropagation algorithm with Adam optimizer, and a maximum number of 60 epochs.

5) PREDICTION OF LANDING POINTS

For a given a saccade sequence, the first prediction was provided from the first 10ms and updated every 5ms using its corresponding model and the samples available to make the prediction. The $(\mathbf{x}^i, \mathbf{y}^i)$ coordinates of the saccade were used as input of the trained model (truncated to the nearest N) and the output of the network was provided as the predicted landing point (x^i_L, y^i_L).

III. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL PROTOCOL

The experimental protocol used to train the models was based on an open-set setup. The participants employed to train the saccade prediction models were not used to evaluate such models. The database was split into training (80% of the participants available in the dataset) and testing set (remaining 20% of the participants). These sets were chosen randomly, and the process was repeated ten times. The performances obtained from all ten experiments were averaged to provide the results. Each user (i.e. individual) contributed to the test set with a different number of samples (i.e. saccades). To avoid a biased result, the prediction error was first calculated individually for each user, and then averaged for all users in the test set.

The experiments include nine different approaches defined by the different learning strategies proposed in Section 2.2, plus three baseline approaches published in previous works [24], [25]:

- *Polynomial Fitting [24]:* this method was one the first approaches of saccade-landing-point prediction in the literature. The authors modelled the amplitude of the saccade as a polynomial problem. The direction was estimated from the samples available for the prediction (angle between first and last saccade sample). The training method was based on a polynomial fitting that minimized the error between the prediction and the real landing point. To guarantee a fair comparison, we applied the proposed approach to the training data available in our dataset.

TABLE 1. Performance of the different approaches (standard deviation in parentheses). The error rates are provided for saccades with length (E_d) under 5 degrees, under 15 degrees, saccades larger than 15 degrees, and average error for all saccades. The errors were calculated as the average of the six errors obtained for the six models trained ($N = 10, 15, \ldots, 35$). Bi = Bidirectional layers, T = Target, E = Average Prediction Error, ET = Averaged Execution Time in milliseconds for a single prediction (input sequence with $N = 10$ coordinates).

Approach	Ref.	Method	T	Bi	Hid. Un.	#Param*	Loss	$E_{d<5}$	$E_{d<15}$	$E_{d>15}$	E_{All}	ET
Baseline	[24]	Polyfit	-	-	-	18	-	$1.87_{(0.13)}$	$2.34_{(0.17)}$	$8.43_{(0.35)}$	$3.62_{(0.21)}$	0.1
A	[25]	FFNN	x_M, y_M	no	32×32	2K	$\mathcal{L}_1$	$1.39_{(0.09)}$	$1.71_{(0.11)}$	$6.50_{(0.31)}$	$1.87_{(0.15)}$	4.5
B	-	RNN	x_M, y_M	no	32×32	2K	$\mathcal{L}_1$	$1.38_{(0.09)}$	$1.70_{(0.10)}$	$6.48_{(0.30)}$	$1.85_{(0.15)}$	4.5
C	[25]	LSTM	x_M, y_M	no	32×32	13K	$\mathcal{L}_1$	$1.37_{(0.08)}$	$1.68_{(0.10)}$	$6.32_{(0.30)}$	$1.71_{(0.14)}$	4.6
D	-	LSTM	α, d	no	32×32	13K	$\mathcal{L}_1$	$1.35_{(0.09)}$	$1.92_{(0.12)}$	$7.58_{(0.42)}$	$2.34_{(0.19)}$	4.6
E	-	LSTM	x_M, y_M	no	32×32	15K	$\mathcal{L}_3$	$1.24_{(0.06)}$	$1.46_{(0.08)}$	$5.66_{(0.26)}$	$1.54_{(0.14)}$	4.8
F	-	LSTM	x_M, y_M	no	64×64	54K	$\mathcal{L}_3$	$1.23_{(0.07)}$	$1.45_{(0.08)}$	$5.57_{(0.26)}$	$1.51_{(0.13)}$	5.2
G	-	LSTM	x_M, y_M	no	128×128	207K	$\mathcal{L}_3$	$1.24_{(0.7)}$	$1.45_{(0.9)}$	$5.49_{(0.25)}$	$1.52_{(0.13)}$	5.3
H	-	LSTM	x_M, y_M	yes	32×32	25K	$\mathcal{L}_3$	$1.29_{(0.7)}$	$1.51_{(0.10)}$	$5.72_{(0.27)}$	$1.65_{(0.15)}$	8.8

*The number of parameters of the model is calculated for input sequences with N=10 coordinates.

- *Approach A - Feed Forward Neural Networks (FFNN) [25]:* FFNNs are the basic architecture of Neural Networks and have been demonstrated to be useful modeling nonlinear functions [52]. This approach was composed of two hidden layers (32 units and ReLu activation of each), and an output layer with two units (i.e. coordinates of the landing point). The $(\bar{\mathbf{x}}^{\mathbf{i}}, \bar{\mathbf{y}}^i)$ coordinates were used as the input of the model.
- *Approach B – RNN:* A RNNs is an architecture trained to model temporal relationships in machine learning problems. The $(\bar{\mathbf{x}}^i, \bar{\mathbf{y}}^i)$ coordinates were used as the input of the model.
- *Approach C – LSTM and Coordinate Regression [25]:* A LSTM is a RNN architecture specifically designed to solve the vanishing problem using memory units. In this approach, the learning strategy was defined by the coordinate target function $\mathbf{T}_1 = (x_M, y_M)$ and the regression loss function $\mathcal{L}_1$.
- *Approach D - LSTM and Direction And Displacement Regression:* the learning strategy of this approach was defined by the target functions ($\mathbf{T}_2 = \alpha$, $\mathbf{T}_3 = d$) and the regression loss function $\mathcal{L}_1$.
- *Approaches E, F, G – LSTM And Fine-Grained Learning:* these approaches were defined by the coordinate target $\mathbf{T}_1 = (x_M, y_M)$ and the combined loss function $\mathcal{L}_3$. The difference between approaches D to F was the number of units in the LSTM layers.
- *Approach H – Bidirectional LSTM and Fine-Grained Learning:* this approach introduced bidirectional learning to approach D. Bidirectional networks connected hidden layers of opposite directions to the same output. These layers allowed us to model previous and next positions with more information.

For each approach, we trained six models ($N = 10, 15, 20, 25, 30, 35$) according to the protocol presented in Section III.A. All approaches were evaluated using the same evaluation set. The error was calculated as the l^2 norm between the predicted and the real landing points.

B. PREDICTION PERFORMANCE

Table 1 presents the performance obtained by the different approaches. The errors were calculated as the Euclidean distance between the real (x_M, y_M) and the predicted landing point (x_L, y_L). All errors were averaged depending on their length: i) saccades shorter than 5 degrees (column 9), ii) saccades shorter than 15 degrees (column 10) iii) saccades longer than 15 degrees (column 11), and iv) all lengths (column 12). Additionally, we report the average prediction error for all the saccades in the test set (in natural viewing, the number of small saccades is much larger than long saccades [11], and see Figure 4b). Table 1 shows the superior performance of our proposed LTSM algorithm (approaches D to G) with average improvement of more than two degrees compared to two state-of-the-art methods [24], [25]. The results show the superior performance of the learning strategy based on predicting the final landing points (approaches E to H) in comparison with the method based on direction and displacement (approach D). The results suggest that the network was capable of modeling the entire movement using the exclusively the initial part of the sequence. Displacement, curvature, and angles were modelled all together in this strategy. Regarding the loss function, the combined loss ($\mathcal{L}_3$) outperformed the regression loss ($\mathcal{L}_1$). The learning process involving both regression and cross-entropy (approaches E to G) was better than the approaches based on traditional regression functions (approaches B and C). Finally, the bidirectional learning

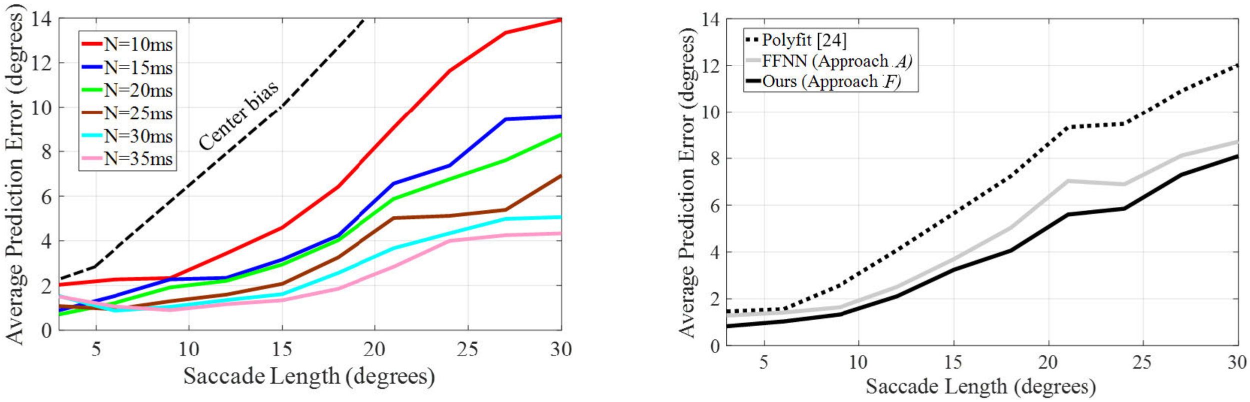

FIGURE 6. Average Prediction Error vs. Saccade Length of our approach *F* according to the number of samples (*N*) used for the prediction (Left). Comparison with previous approaches (Right). The performance of the approaches are shown as unique curves (Right) that were calculated as the average of the six curves ($N = 10, 15, \ldots, 35$). As a baseline, center bias was obtained calculating predictions by sampling randomly from a 2D spatial Gaussian distribution whose mean and variance were determined according to the center bias model [53].

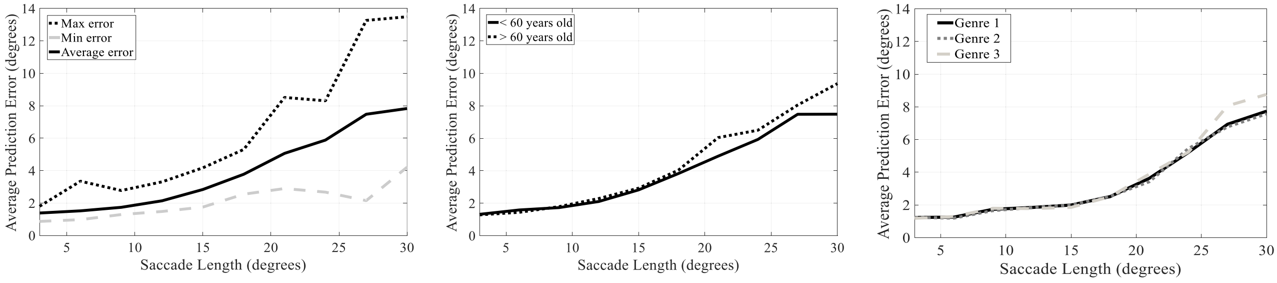

FIGURE 7. Average Prediction Error vs. Saccade Length of the best approach (approach *F*) according to the number of samples (*N*) used for the prediction. Maximum, minimum, and average Prediction Error obtained for the different users (Left). Prediction error for users younger and older than 60 years old (Center). Prediction error obtained for videos with different Genre (Genre 1 = Nature documentary, Genre 2 = Cartoon, Genre 3 = Drama) (Right). The performance of the approaches are shown as unique curves that were calculated as the average of the six curves ($N = 10, 15, \ldots, 35$).

(approach *D*) did not improve the forward learning (approach *E*). Table 1 includes the standard deviation of the errors. Additionally, Table 1 summarizes the number of parameters of the approaches evaluated in this work. While the polynomial method proposed in [24] was defined by 18 parameters, the neural network architectures were comprised of many more parameters that could be successfully trained for saccade-landing-point prediction due to the large number of saccades available in the datasets that are becoming available.

Table 1 includes the execution times for a prediction of an input saccade of ten milliseconds ($N = 10$). All the experiments were carried out in an Intel Core i7-8750H CPU @ 2.20GHz, 32 GB RAM, Nvidia GeForce RTX 2080. As was expected, the polynomial fitting had the lowest execution time. Deep learning approaches were around 5 milliseconds with small differences between approaches, except for the Bidirectional layer with an execution time of 8.8 milliseconds. Note that execution time of the deep learning approaches varied with the input size N. For example, the execution time for different input sizes ($N = 10, 15, 20, 25, 30, 35$) of the approach F were [5.2, 6.1, 7.3, 8.2, 9.4, 10.4 milliseconds]. Note that these execution times could be improved with dedicated hardware or subsampled input sequences.

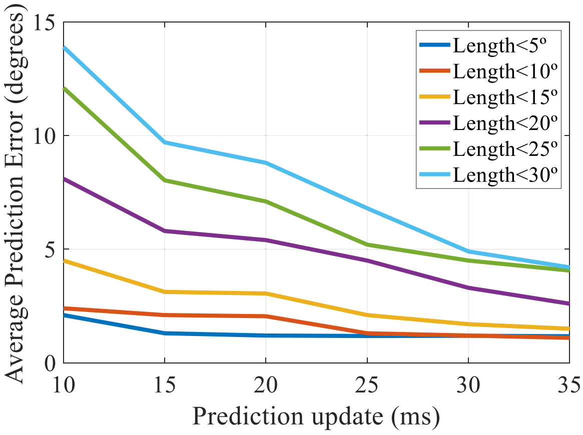

FIGURE 8. Average Prediction Error vs. Prediction update (ms), approach *F*, according to the length of the saccade (Colored image).

For the rest of the experiments, we used model F as it had a good balance between prediction accuracy and number of parameters. The prediction error was inversely proportional

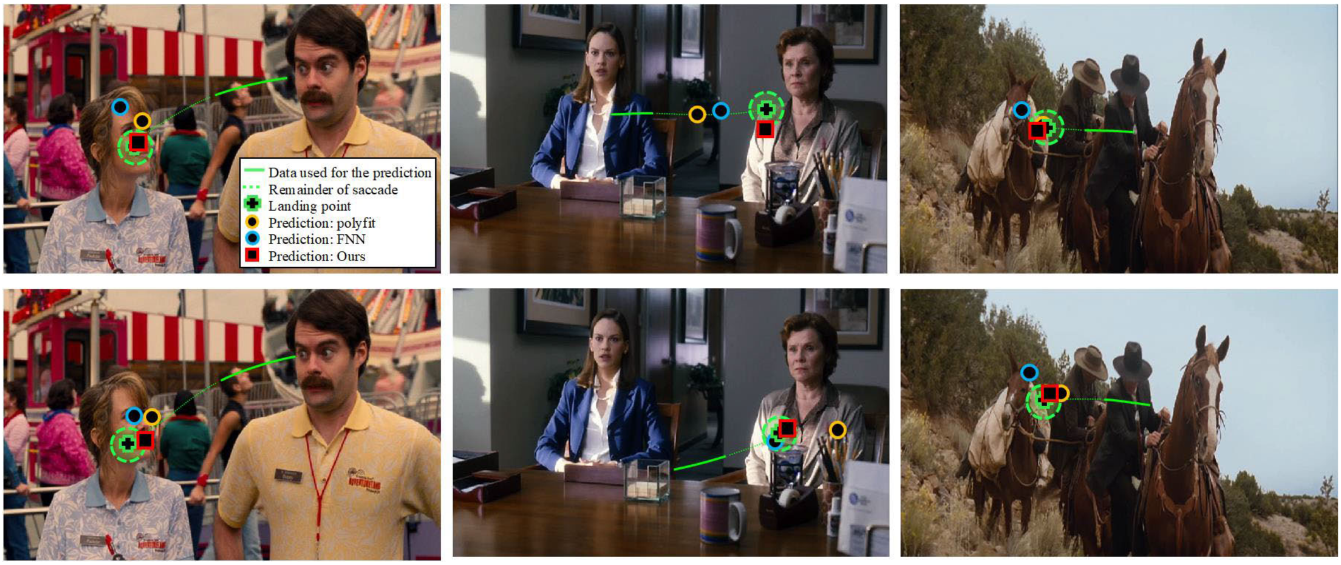

FIGURE 9. **Example of the performance of the proposed method, approach *F*, and two previous approaches over real saccade trajectories. The two rows show saccades for two different participants. The trajectories have been smoothed for visualization purposes. Number of samples used for the prediction $N = 15$. The gaze positions are in the region of interest of the viewer, with the fovea represented by the green circles, each with a diameter of 2.5° (Colored image).**

to the number of samples (N) used to make the prediction. Figure 6 (Left) shows how the performance of the proposed approach depended on N. The results show how the prediction error decreased when more information was available for the prediction. As described in Section 2.D, the landing point prediction was updated once new samples were available. Thus, the error decreased as the saccade progressed. This update handled the non-linear trajectories of long saccades that are common [11]. The error varied from less than 1 degree for the shortest saccades to 15 degrees for predictions made with only 10ms and the largest saccades (30 degrees). In Figure 6 (Right) we show how the proposed approach outperformed previous approaches for all saccade lengths. The fine-grained learning was better than the polynomial fitting approach [24] by more than two degrees, on average. In comparison to the FFNN proposed in [25], there was an improvement of 0.36 degrees, on average.

C. EVALUATION OF THE EFFECTS OF COVARIATES

Landing point predictions were affected by many different factors including user-dependent (e.g. neuromotor physiognomy) and task-dependent (e.g. number of samples available to make the prediction, length of the saccade) characteristics.

Some user characteristics affected the performance of the prediction. Figure 7 (Left) shows the average performance and that obtained by for best and worst participants. The large between-subject difference between the best and worst participants was significant (linear regression, $t = 8.79$, $p < 0.001$), especially for the largest saccades. We found that subject age was related to the quality of the prediction. Figure 7 (Center) shows that prediction was better for younger than older participants (shown with 60 years criterion). Presumably, older age affected the neuromotor control of the participants, and thus increasing age affected prediction performance (older subjects had slightly higher errors than younger subjects: $t = 2.23$, $p = 0.04$). Figure 7 (Right) shows that there was no difference between the genres ($F(2, 20) = 0.17$, $p = 0.84$).

Visual stimuli are the primary drivers of visual attention and thence saccades to new regions of interest. The stimuli determine the length and direction of the saccades and therefore, they affect the accuracy of the predictions. The type and salience of visual stimuli might vary between video genres. Figure 7 (Right) shows the prediction error for different types of genres (Nature Documentary, Cartoon, and Drama). Our results suggest that the genre had a limited impact on the performance of our predictions (p-value $= 0.98$). We are not claiming that visual stimuli do not affect the performance of our approach. While visual stimuli are critical to the length of the saccade, with equal length, the prediction errors were similar for the three genres.

Figure 8 shows the performance when the landing point was updated with data provided by the eye tracker (i.e. in-flight update). In this experiment, the landing point prediction was updated every 5 milliseconds. Figure 8 shows how the predictions improved with the time and reduced the errors. This in-flight updating is an advantage of online landing prediction approaches based on eye tracking data.

To illustrate the predictions of saccade-landing location, Figure 9 shows a comparison of the predictions made by three different approaches during real natural viewing for two participants watching each of three video clips. These predictions of real saccades reveal the suitability of this technology for real-world applications.

IV. CONCLUSION

Saccades are rapid eye movements that define the region of interest of a viewer. The prediction of the saccade landing point allows a partial solution to the problems related to the update latency of gaze-contingent displays. We propose a novel algorithm to train saccade landing point predictor based on LSTM networks and fine-grained learning approach. Our strategy includes a dual loss function trained according to both gross and fine predictions. We evaluated 8 different approaches over 220,000 sequences obtained from 75 users. The data used to train and evaluate the approaches was acquired monitoring the eye movements of the participants during natural viewing of videos.

The results showed the superior performance of our proposed method with average improvement of more than two degrees in comparison with state-of-the-art methods [24], [25]. Figure 6 shows a comparison of the predictions made by different approaches during real natural viewing (Fig. 6 Right). We presented a comprehensive analysis of the performance including experiments that varied the number of samples used for the prediction (Fig. 6 Left), user-dependent effects (Fig. 7 Left), variation due to the age of the participant (Fig. 7 Center), and different video genres (Fig. 7 Right). Saccadic length and user characteristics showed the largest impact on the performance. The errors varied depending on the saccade length, with errors under three degrees for saccades smaller than 15 degrees, and errors around eight degrees for the largest saccades. Differences between participants in average prediction error was up to 9.5 degrees for the largest saccades.

Future works will include the study of user-dependent models that allow capturing the intra-user variability. The LSTM networks trained in this work could be tuned for each subject (if there was enough information from the subject) in a similar way to other domain-adaptation methods and transfer-learning techniques. The proposed learning approach can be used to predict the duration of the saccade, which is a useful information in gaze-contingent rendering. Other future research lines include the study of multimodal approaches developed to exploit both visual stimuli [12], [45]–[47] and eye tracker data [22], [24]–[26].

REFERENCES

[1] S. M. Reder, "On-line monitoring of eye-position signals in contingent and noncontingent paradigms," *Behav. Res. Methods Instrum.*, vol. 5, no. 2, pp. 218–228, Mar. 1973.

[2] A. T. Duchowski, N. Cournia, and H. Murphy, "Gaze-contingent displays: A review," *CyberPsychology Behav.*, vol. 7, no. 6, pp. 621–634, Dec. 2004.

[3] E. M. Reingold, L. C. Loschky, G. W. McConkie, and D. M. Stampe, "Gaze-contingent multiresolutional displays: An integrative review," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 45, no. 2, pp. 307–328, Jun. 2003.

[4] G. P. Mylonas, K.-W. Kwok, D. R. C. James, D. Leff, F. Orihuela-Espina, A. Darzi, and G.-Z. Yang, "Gaze-contingent motor channelling, haptic constraints and associated cognitive demand for robotic MIS," *Med. Image Anal.*, vol. 16, no. 3, pp. 612–631, Apr. 2012.

[5] J. Wade, L. Zhang, D. Bian, J. Fan, A. Swanson, A. Weitlauf, M. Sarkar, Z. Warren, and N. Sarkar, "A gaze-contingent adaptive virtual reality driving environment for intervention in individuals with autism spectrum disorders," *ACM Trans. Interact. Intell. Syst.*, vol. 6, no. 1, pp. 1–23, May 2016.

[6] M. R. Vasilev and B. Angele, "Parafoveal preview effects from word $N+$ 1 and word $N+$ 2 during reading: A critical review and Bayesian meta-analysis," *Psychonomic Bull. Rev.*, vol. 24, no. 3, pp. 666–689, 2017.

[7] J. M. Henderson, K. K. Mcclure, S. Pierce, and G. Schrock, "Object identification without foveal vision: Evidence from an artificial scotoma paradigm," *Perception Psychophysics*, vol. 59, no. 3, pp. 323–346, Apr. 1997.

[8] F. M. Costela, S. M. Reeves, and R. L. Woods, "Orientation of the preferred retinal locus (PRL) is maintained following changes in simulated scotoma size," *J. Vis.*, vol. 20, no. 7, p. 25, Jul. 2020.

[9] F. Santini, G. Redner, R. Iovin, and M. Rucci, "EyeRIS: A general-purpose system for eye-movement-contingent display control," *Behav. Res. Methods*, vol. 39, no. 3, pp. 350–364, Aug. 2007.

[10] D. R. Saunders and R. L. Woods, "Direct measurement of the system latency of gaze-contingent displays," *Behav. Res. Methods*, vol. 46, no. 2, pp. 439–447, Jun. 2014.

[11] F. M. Costela and R. L. Woods, "When watching video, many saccades are curved and deviate from a velocity profile model," *Frontiers Neurosci.*, vol. 12, p. 960, Jan. 2019.

[12] A. Borji, "Saliency prediction in the deep learning era: Successes and limitations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 679–700, Feb. 2021.

[13] A. T. Bahill, M. R. Clark, and L. Stark, "The main sequence, a tool for studying human eye movements," *Math. Biosci.*, vol. 24, nos. 3–4, pp. 191–204, Jan. 1975.

[14] J. Anliker, "Eye movements: Online measurement, analysis, and control," in *Eye Movements and Psychological Processes*, R. A. Monty and J. W. Senders, Eds. Hillsdale, NJ, USA: Lawrence Erlbaum, 1976, pp. 185–202.

[15] P. Han, D. R. Saunders, R. L. Woods, and G. Luo, "Trajectory prediction of saccadic eye movements using a compressed exponential model," *J. Vis.*, vol. 13, no. 8, p. 27, Jul. 2013.

[16] C. Paeye, A. C. Schütz, and K. R. Gegenfurtner, "Visual reinforcement shapes eye movements in visual search," *J. Vis.*, vol. 16, no. 10, p. 15, Aug. 2016.

[17] S. H. Yeo, M. Lesmana, D. R. Neog, and D. K. Pai, "Eyecatch: Simulating visuomotor coordination for object interception," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–10, Aug. 2012.

[18] O. V. Komogortsev and J. I. Khan, "Eye movement prediction by oculomotor plant Kalman filter with brainstem control," *J. Control Theory Appl.*, vol. 7, no. 1, pp. 14–22, Feb. 2009.

[19] O. V. Komogortsev, Y. S. Ryu, D. H. Koh, and S. M. Gowda, "Instantaneous saccade driven eye gaze interaction," in *Proc. Int. Conf. Adv. Comput. Enterntainment Technol. (ACE)*, Athens, Greece, 2009, pp. 140–147.

[20] O. V. Komogortsev, Y. S. Ryu, and D. H. Koh, "Quick models for saccade amplitude prediction," *J. Eye Movement Res.*, vol. 3, no. 1, pp. 1–13, Jun. 2009.

[21] S. Wang, R. L. Woods, F. M. Costela, and G. Luo, "Dynamic gaze-position prediction of saccadic eye movements using a Taylor series," *J. Vis.*, vol. 17, no. 14, p. 3, Dec. 2017.

[22] H. Griffith, S. Biswas, and O. Komogortsev, "Towards reduced latency in saccade landing position prediction using velocity profile methods," in *Proc. Future Technol. Conf.*, Vancouver, BC, Canada, 2018, pp. 79–91.

[23] A. J. Van Opstal and J. A. M. Van Gisbergen, "Skewness of saccadic velocity profiles: A unifying parameter for normal and slow saccades," *Vis. Res.*, vol. 27, no. 5, pp. 731–745, Jan. 1987.

[24] E. Arabadzhiyska, O. T. Tursun, K. Myszkowski, H.-P. Seidel, and P. Didyk, "Saccade landing position prediction for gaze-contingent rendering," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Jul. 2017.

[25] A. Morales, F. M. Costela, R. Tolosana, and R. L. Woods, "Saccade landing point prediction: A novel approach based on recurrent neural networks," in *Proc. Int. Conf. Mach. Learn. Technol. (ICMLT)*, Jinan, China, 2018, pp. 1–5.

[26] H. Griffith and O. Komogortsev, "A shift-based data augmentation strategy for improving saccade landing point prediction," in *Proc. Symp. Eye Tracking Res. Appl.*, Stuttgart, Germany, Jun. 2020, pp. 1–6.

[27] E. Castet, "Perception of intra-saccadic motion," in *Dynamics of Visual Motion Processing: Neuronal, Behavioral, and Computational Approaches*, U. J. Ilg and G. S. Masson, Eds. New York, NY, USA: Springer, 2010.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2921–2929.

[30] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[31] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 1–9.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[33] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[34] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "Exploring recurrent neural networks for on-line handwritten signature biometrics," *IEEE Access*, vol. 6, pp. 5128–5138, 2018.

[35] A. Graves, A. R. Mohamed, and G. Hinton, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, vol. 14, 2014, pp. 1764–1772.

[36] A. Petrosian, D. Prokhorov, R. Homan, R. Dasheiff, and D. Wunsch, "Recurrent neural network based prediction of epileptic seizures in intra- and extracranial EEG," *Neurocomputing*, vol. 30, nos. 1–4, pp. 201–218, Jan. 2000.

[37] S. Hochreiter, Y. Bengio, and P. J. F. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks*, S. C. Kremer and J. F. Kolen, Eds. Hoboken, NJ, USA: Wiley, 2001.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 321–328.

[40] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Salt Lake City, UT, USA, Jun. 2018, pp. 2074–2083.

[41] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Proc. Annu. Conf. Neural Inf. Process Syst.*, Montreal, QC, Canada, 2005, pp. 155–162.

[42] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 989–1005, Jun. 2009.

[43] C. Xia, F. Qi, and G. Shi, "Bottom–up visual saliency estimation with deep autoencoder-based sparse reconstruction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1227–1240, Jun. 2016.

[44] X. Sun, H. Yao, R. Ji, and X.-M. Liu, "Toward statistical modeling of saccadic eye-movement and visual saliency," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4649–4662, Nov. 2014.

[45] C. Xia, J. Han, F. Qi, and G. Shi, "Predicting human saccadic scanpaths based on iterative representation learning," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3502–3515, Jul. 2019.

[46] W. Wang, C. Chen, Y. Wang, T. Jiang, F. Fang, and Y. Yao, "Simulating human saccadic scanpaths on natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Colorado Springs, CO, USA, Jun. 2011, pp. 441–448.

[47] O. Le Meur, A. Coutrot, Z. Liu, P. Rämä, A. Le Roch, and A. Helo, "Visual attention saccadic models learn to emulate gaze patterns from childhood to adulthood," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4777–4789, Oct. 2017.

[48] F. M. Costela and R. L. Woods, "The impact of field of view on understanding of a movie is reduced by magnifying around the center of interest," *Transl. Vis. Sci. Technol.*, vol. 9, no. 8, p. 6, Jul. 2020.

[49] F. M. Costela and R. L. Woods, "A free database of eye movements watching 'Hollywood' video clips," *Data Brief*, vol. 25, Aug. 2019, Art. no. 103991.

[50] F. M. Costela, S. Kajtezovic, and R. L. Woods, "The preferred retinal locus used to watch videos," *Investigative Ophthalmol. Vis. Sci.*, vol. 58, no. 14, pp. 6073–6081, 2017.

[51] D. R. Saunders, P. J. Bex, and R. L. Woods, "Crowdsourcing a normative natural language dataset: A comparison of Amazon Mechanical Turk and in-lab data collection," *J. Med. Internet Res.*, vol. 15, no. 5, p. e100, May 2013.

[52] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[53] P. H. Tseng, R. Carmi, I. G. M. Cameron, D. P. Munoz, and L. Itti, "Quantifying center bias of observers in free viewing of dynamic natural scenes," *J. Vis.*, vol. 9, no. 7, p. 4, Jul. 2009.

AYTHAMI MORALES received the M.Sc. degree in electrical engineering and the Ph.D. degree in artificial intelligence from ULPGC, in 2006 and 2011, respectively. Since 2017, he has been an Associate Professor with UAM. He has conducted research stays at Michigan State University, Hong Kong Polytechnic University, the University of Bologna, and the Schepens Eye Research Institute. He has authored over 100 scientific articles in topics related to machine learning, trustworthy AI, and biometric signal processing.

FRANCISCO M. COSTELA received the M.Sc. degree in biomedical engineering and neuroscience from Oxford University, in 2009, and the M.Sc. degree in biomedical engineering and neuroscience from the Pablo de Olavide University, in 2010, and the Ph.D. degree in visual neuroscience from the Barrow Neurological Institute, Arizona State University, in 2014. He became proficient on vision sciences, computer vision, and machine learning, authoring over 25 scientific contributions in related topics.

RUSSELL L. WOODS is currently with Schepens Eye Research Institute, Massachusetts Eye and Ear, where his research primarily focuses on vision impairment and rehabilitation. He is also developing and evaluating methods to modify the appearance of video from television, movies, and other formats. He also seeks to understand the visual input to pedestrian mobility and evaluates devices that help restore mobility that is lost due to vision impairment. Recent novel methods include improvements to gaze-contingent display systems and a method to assess the acquisition of sensory information (e.g., can you understand what you see?). He is also developing a method to measure cognitive impairments and testing a novel method of treating visuo-spatial neglect.