Neuro-inspired Eye Tracking with Eye Movement Dynamics

Kang Wang
RPI
kangwang.kw@gmail.com

Hui Su
RPI and IBM
huisuibmres@us.ibm.com

Qiang Ji
RPI
qji@ecse.rpi.edu

Abstract

Generalizing eye tracking to new subjects/environments remains challenging for existing appearance-based methods. To address this issue, we propose to leverage on eye movement dynamics inspired by neurological studies. Studies show that there exist several common eye movement types, independent of viewing contents and subjects, such as fixation, saccade, and smooth pursuits. Incorporating generic eye movement dynamics can therefore improve the generalization capabilities. In particular, we propose a novel Dynamic Gaze Transition Network (DGTN) to capture the underlying eye movement dynamics and serve as the top-down gaze prior. Combined with the bottom-up gaze measurements from the deep convolutional neural network, our method achieves better performance for both within-dataset and cross-dataset evaluations compared to state-of-the-art. In addition, a new DynamicGaze dataset is also constructed to study eye movement dynamics and eye gaze estimation.

1. Introduction

Eye gaze is one of the most important approaches for people to interact with each other and with the visual world. Eye tracking has been applied to different fields, including psychology study [1], social network [2, 3, 4, 5], web search [6, 7, 8], marketing and advertising [9], human computer interaction [10, 11, 12]. In addition, since neurological activities affect the way to process visual information (reflected by eye movements), eye tracking, therefore becomes one of the most effective tools to study neuroscience. The estimated eye movements, eye gaze patterns can help attentional studies like object-search mechanisms [6], understand neurological functions during perceptual decision making [13], and medical diagnosis like schizophrenia, post-concussive syndrome, autism, Fragile X, etc. Despite the importance of eye tracking to neuroscience studies, researchers ignored that neurological studies on eyes can also benefit eye tracking. It is revealed that eye tracking is not a random process but involves strong dynamics. There exist common eye movement dynamics [1] that are independent of the viewing content and subjects. Exploiting eye movement dynamics can significantly improve the performance of eye tracking.

From neuroanatomy studies, there are several major types of eye movements [2]: vergence, saccade, fixation and smooth pursuit. Vergence movements are to fixate on objects at different distances where two eyes move in opposite direction. As vergence is less common in natural viewing scenarios, we mainly focus on fixation, saccade, and smooth pursuit eye movements. Saccadic movement is rapid eye movement from one fixation to another, its duration is short and the amplitude is linearly correlated with the duration. There are also study on microsaccade [14] which is not the focus of this paper. Fixation is to fixate on the same object for a period of time, eye movements are very small (miniature) and can be considered as a stationary or random walk. Smooth pursuit is eye movement which smoothly tracks a slowly moving object. It cannot be triggered voluntarily and typically require a moving object.

Existing work (see [15] for a comprehensive survey) on eye gaze estimation are static frame-based, without explicitly considering the underlying dynamics. Among them, model-based methods [16, 17, 18, 19, 20, 21, 22, 23, 24, 25] estimate eye gaze based on a geometric 3D eye model. Eye gaze can be estimated by detecting key points in the geometric 3D eye model. Differently, appearance-based methods [26, 27, 28, 29, 30, 31] directly learn a mapping function from eye appearance to eye gaze.

Unlike traditional static frame-based methods, we propose to estimate eye gaze with the help of eye movement dynamics. Since eye movement dynamics can generalize across subjects and environments, the proposed method therefore achieves better generalization capabilities. The system is illustrated in Fig. 1. For online eye tracking, the static gaze estimation network first estimates the raw gaze $\mathbf{x}_t$ from input frame. Next, we combine top-down eye movement dynamics with bottom-up image measurements (Alg. 1) to get a more accurate prediction $\mathbf{y}_t$. In addition, $\mathbf{y}_t$ is further fed back to refine the static network so that we can better generalize to

[1] In this work, eye movement refers to actual gaze movement on screens.
[2] https://www.ncbi.nlm.nih.gov/books/NBK10991/

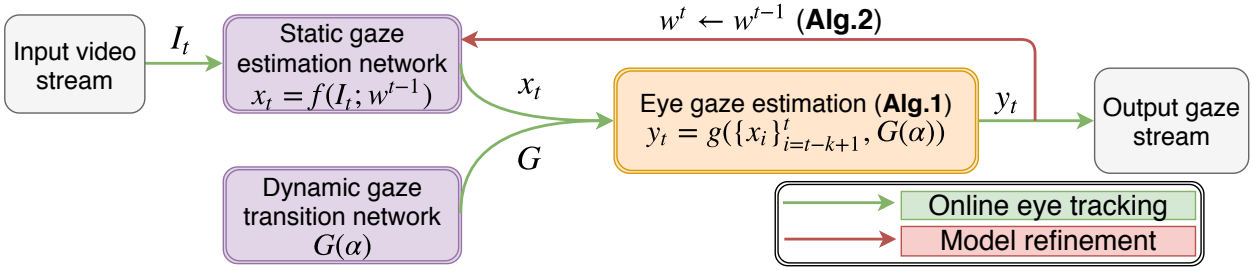


Figure 1. Overview of the proposed system. For online eye tracking, we combine static gaze estimation network with dynamic gaze transition network to obtain better gaze estimation. In addition, the feedback mechanism of the system allows model refinement, so that we can better generalize the static network to unseen subjects or environments.

current user and environment (Alg. 2). The proposed method makes following contributions:

- To the best of our knowledge, we are the first to take advantage of dynamic information to improve gaze estimation. Combining top-down eye movement dynamics with bottom-up image measurements gives better generalization and accuracy (%15 improvement), and can automatically adapt to unseen subjects and environments.
- Propose the DGTN that effectively captures the transitions of different eye movements as well as their underlying dynamics.
- Construct the DynamicGaze dataset, which not only provides another benchmark for evaluating static gaze estimation but benefits the community for studying eye gaze and eye movement dynamics.

2. Related Work

Static eye gaze estimation. The most relevant work to our static gaze estimation is from [27]. The authors proposed to estimate gaze on mobile devices with face, eye and head pose information using a deep convolutional neural network. Though they can achieve good performance within-dataset, they cannot generalize well to other datasets.

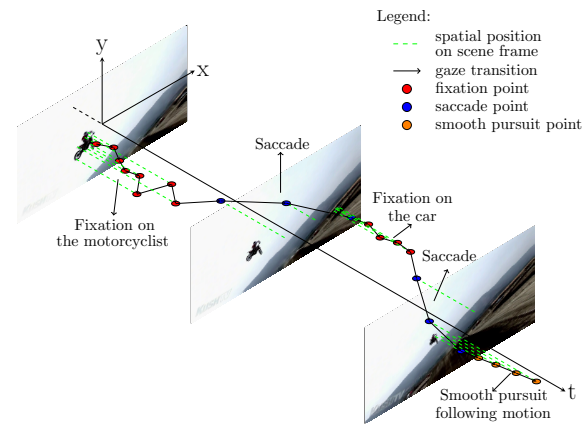
Eye gaze estimation with eye movement dynamics. Eye movement is a spatial-temporal process. Most existing work only uses spatial eye movements, also known as saliency map. In [32, 18, 33], the authors approximated the spatial gaze distribution with the saliency map extracted from image/video stimulus. However, their purpose is to perform implicit personal calibration instead of improving gaze estimation accuracy, since spatial saliency map is scene-dependent. In [34], the authors used the fact that over 80% chance that first two fixations are on faces to help estimate eye gaze. However, their approximation is too simple and cannot apply to more natural scenarios.

For temporal eye movements, the authors in [35] proposed to estimate the future gaze positions for recommender systems with a Hidden Markov Model (HMM), where fixation is assumed to be a latent state, and user actions (clicking, rating, dwell time, etc) are the observations. Their method is however very much task-dependent and cannot generalize to different tasks. In [36], the authors proposed to use a similar HMM to predict gaze positions to reduce the delay of networked video streaming. They also considered three states corresponding to fixation, saccade, and smooth pursuit. However, their approach ignores the different duration for the three states, and their detailed modeling of the dynamics for each state is relatively simpler. In addition, it requires a commercial eye tracker, while the proposed method is an appearance-based gaze estimator, which can perform online real-time eye tracking with a simple web-camera. Furthermore, the proposed method supports model-refinement which can generalize to new subjects and environments.

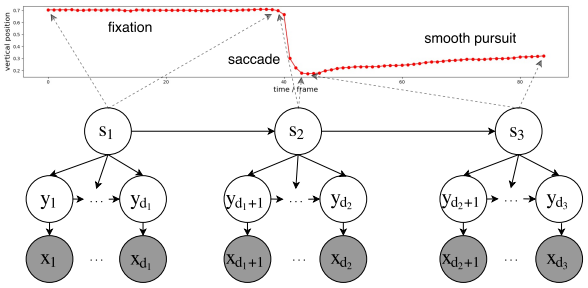
Eye Movement Analysis. Besides eye tracking, there are plenty of work on identifying the eye movement types given eye tracking data. It includes threshold-based [37, 38] and probabilistic-based [39, 40, 41]. Both methods require measurements from eye tracking data like dispersion, velocity or acceleration. Analyzing the underlying distribution of these measurements can help identify the eye movement types. However, these approaches are not interested in modeling the gaze transitions for improving eye tracking.

3. Proposed Framework

We first discuss the eye movement dynamics and the DGTN in Sec. 3.1. Next, we briefly introduce the static gaze estimation network in Sec. 3.2. Then we talk about how to perform online eye tracking with top-down eye movement dynamics and bottom-up gaze measurements in Sec. 3.3. Finally in Sec. 3.4, we focus on the refinement of the static gaze estimation network.



(a)



(b)

Figure 2. Eye movement dynamics. (a) Illustration of eye movements while watching a video, (b) Graphical representation of dynamic gaze transition network.

3.1. Eye Movement Dynamics and DGTN

We first take a look at the eye movements while watching a video. As shown in Fig. 2 (a), the user is first attracted by the motorcyclist on the sky. After spending some time fixating on the motorcyclist, the user shifts the focus on the recently appeared car (due to shooting angle change). A saccade is in between of the two fixations. Next, the user turns the focus back to the motorcyclist and starts following the motion with smooth pursuit eye movement. We have three observations regarding the eye movements: 1) each eye movement has its own unique dynamic pattern, 2) different eye movements have different durations, and 3) there exists special transition patterns across different eye movements. These observations inspire us to construct the dynamic model shown in Fig. 2 (b) to model the overall gaze transitions.

Specifically, we employ the semi-Markov model to model the durations for each eye movement type. In Fig. 2 (b), the red curve on the top shows a sample gaze pattern with 3 segments of fixation, saccade, and smooth pursuit respectively. The top row represents the state chain s_t, where $s_t = \{fix, sac, sp\}$ can take three values corresponding to fixation, saccade, and smooth pursuit respectively. Each state can generate a sequence of true gaze positions $\{\mathbf{y}_t\}_{t=1}^{d}$, where d represents the duration for the state. Though the state s_t is constant for a long period, its value is copied for all time slices within the state to ensure a regular structure. The true gaze $\mathbf{y}_t$ not only depends on the current state but also depends on previous gaze positions. For example, the moving direction for smooth pursuit is determined by several previous gaze positions. Given the true gaze $\mathbf{y}_t$, we can generate the noisy measurements $\mathbf{x}_t$, which are the outputs from the static gaze estimation methods.

In the following, we will discuss in details 1) within-state dynamics (Sec. 3.1.1), 2) eye movement duration and transition (Sec. 3.1.2), 3) measurement model (Sec. 3.1.3), and 4) parameter learning (Sec. 3.1.4).

3.1.1 Within-state Dynamics

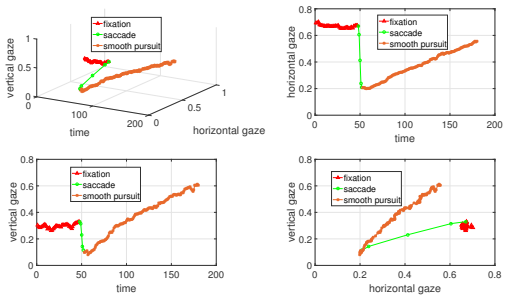
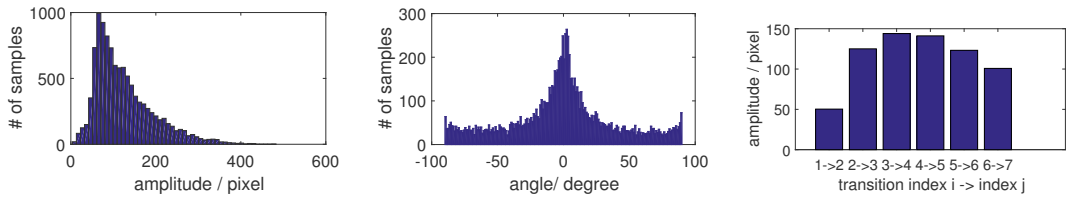


Figure 3. Visualization of eye movements. top-left: 3D plot of x-y-t; top-right: projected 2D plot on y-t plane; bottom-left: projected 2D plot on x-t plane; bottom-right: projected 2D plot on x-y plane.

Fixation. Fixation is to fixate eye gaze on the same static object for a period of time (Fig. 3 (d)). We propose to model it with random walk : $\mathbf{y}_t = \mathbf{y}_{t-1} + \mathbf{w}_{fix}$, where $\mathbf{w}_{fix}$ is the Gaussian noise with zero-mean and covariance matrix of $\mathbf{\Sigma}_{fix}$.

Saccade. Typically, saccade is fast eye movement between two fixations. The trajectory is typically a straight line or generalized exponential curves (Fig. 3). In this work, we approximate the trajectory with piece-wise linear functions. The first saccade point $\mathbf{y}_1$ is actually the end point of last fixation. Predicting the position of second saccade point $\mathbf{y}_2$ is difficult without knowing the image content. However, according to [42], horizontal saccades are more frequent than vertical saccades, which provide strong cues to the second saccade point. Specifically, we assume second point can be estimated by transiting first point with certain amplitude and direction (angle) on 2D plane: $\mathbf{y}_2 = \mathbf{y}_1 + \lambda[\cos(\theta), \sin(\theta)]^T$, where amplitude $\lambda \sim \mathcal{N}(\mu_\lambda, \sigma_\lambda)$ and angle $\theta \sim \mathcal{N}(\mu_\theta, \sigma_\theta)$ both follow Gaussian distributions. The histogram plot of amplitude (Fig. 4 (a)) and angle (Fig. 4 (b)) from real data also validates the feasibility of Gaussian distributions.



(a) (b) (c)

Figure 4. Saccade characteristics. (a) Amplitude distribution, (b) Angle distribution, (c) Amplitude change from adjacent saccade points.

The rest saccade points can be estimated with the previous two points: $\mathbf{y}_t = \mathbf{B}_1^i \mathbf{y}_{t-1} + \mathbf{B}_2^i \mathbf{y}_{t-2} + \mathbf{w}_{sac}$, where $\mathbf{B}_1^d$ and $\mathbf{B}_2^i$ are the regression matrices, the superscript i indicates the index of current saccade point, or how many frames have past when we enter the state. The value of i equals the duration variable d in Eq. (1). It might be easier if we assume $\mathbf{B}_1^i$ and $\mathbf{B}_2^i$ remain the same for different indexes i, but saccade movements have certain characteristics. For example as in (Fig. 4 (c)), the amplitude changes between adjacent saccade points first increases than decreases. Using index-dependent regression matrices can better capture the underlying dynamics. $\mathbf{w}_{sac}$ is the Gaussian noise with zero-mean and covariance matrix of $\mathbf{\Sigma}_{sac}$.

Smooth Pursuit. Smooth pursuit is to keep track of a slowly moving object. Therefore we can approximate the moving trajectory by piece-wise linear functions similar to saccade points. For the second smooth pursuit point, we introduce amplitude and angle variable $\{\lambda_{sp}, \theta_{sp}\}$. For remaining smooth pursuit points, we introduce index-dependent regression matrices: $\mathbf{y}_t = \mathbf{C}_1^i \mathbf{y}_{t-1} + \mathbf{C}_2^i \mathbf{y}_{t-2} + \mathbf{w}_{sp}$. $\mathbf{w}_{sp}$ is the Gaussian noise with zero-mean and covariance matrix of $\mathbf{\Sigma}_{sp}$.

3.1.2 Eye Movement Duration and Transition

The hidden semi-Markov model has been well studied in [43], we adopt a similar formulation for our model in terms of state duration and transition modeling. Besides random variables s_t, $\mathbf{y}_t$ and $\mathbf{x}_t$ for state, true gaze position and measured gaze position, we introduce another discrete random variable d_t (range $\{0, 1, ..., D\}$) representing the remaining duration of state s_t. The state s_t and the remaining duration d_t are discrete random variables and follows multinomial (categorical) distribution. The CPDs for the state transition are defined as follows:

$$P(s_t = j | s_{t-1} = i, d_t = d) = \begin{cases} \delta(i,j) \text{ if } d > 0 \\ \mathbf{A}(i,j) \text{ if } d = 0 \end{cases}$$
$$P(d_t = d' | d_t = d, s_t = k) = \begin{cases} \delta(d', d-1) \text{ if } d > 0 \\ p_k(d') \text{ if } d = 0 \end{cases} \quad (1)$$

where $\delta(i,j) = 1$ if $i = j$ else 0. When we enter a new state $s_t = i$, the duration d_t is drawn from a prior multinomial distribution $q_i(\cdot) = [p_i(1), ..., p_i(D)]$. The duration is then counts down to 0. When $d_t = 0$, the state transits to a different state with the state transition matrix $\mathbf{A}$ and the duration for the new state is drawn again from $q_i(\cdot)$.

3.1.3 Measurement Model

The measurement model $P(\mathbf{x}_t | \mathbf{y}_t)$ is independent of the type of eye movement, and we assume: $\mathbf{x}_t = \mathbf{D}\mathbf{y}_t + \mathbf{w}_n$, where $\mathbf{D}$ is the regression matrix, and $\mathbf{w}_n$ is multi-variate Gaussian noise with zero-mean and covariance matrix of $\mathbf{\Sigma}_n$.

3.1.4 Parameter Learning

The DGTN parameters are summarized in Table 1. For simplicity, we denote all the parameters as $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_{st}, \boldsymbol{\alpha}_{sd}, \boldsymbol{\alpha}_{fix}, \boldsymbol{\alpha}_{sac}, \boldsymbol{\alpha}_{sp}, \boldsymbol{\alpha}_m]$ and the DGTN is represented as $G(\boldsymbol{\alpha})$. All the random variables in Fig. 2 (b) are observed during learning (the states and true gaze are not known during online gaze tracking). Given the fully observed K sequences $(\{s_t^k, \mathbf{y}_t^k, \mathbf{x}_t^k\}_{t=1}^{T_k})$ each with length T_k, we can use Maximum log likelihood to estimate all the parameters:

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} \log \prod_{k=1}^{K} P(\{s_t^k, \mathbf{y}_t^k, \mathbf{x}_t^k\}_{t=1}^{T_k} | \boldsymbol{\alpha}) \quad (2)$$
$$= \arg\max_{\boldsymbol{\alpha}} \sum_{k=1}^{K} \log \prod_{t=1}^{T_k} \sum_{d_t^k} P(s_t^k, d_t^k) P(\mathbf{y}_t^k | s_t^k, d_t^k) P(\mathbf{x}_t^k | \mathbf{y}_t^k)$$

With fully-observed data, the above optimization problem can be factorized to following sub-problems, each of which can be solved independently:

$$\alpha_m^* = \arg\max_{\boldsymbol{\alpha}_m} \sum_{k=1}^{K} \log \prod_{t=1}^{T_k} P(\mathbf{x}_t^k | \mathbf{y}_t^k, \boldsymbol{\alpha}_m), \quad (3)$$

$$\{\boldsymbol{\alpha}_{st}, \boldsymbol{\alpha}_{sd}\}^* = \arg\max_{\boldsymbol{\alpha}_{st}, \boldsymbol{\alpha}_{sd}} \sum_{k=1}^{K} \log \prod_{t=1}^{T_k} \sum_{d_t^k} P(s_t^k, d_t^k) \quad (4)$$

$$\boldsymbol{\alpha}_j^* = \arg\max_{\boldsymbol{\alpha}_j} \sum_{n=1}^{N_j} \log \prod_{t=1}^{T_n} P(\mathbf{y}_t^k | s_t^k = j, d_t^k = T_n, \boldsymbol{\alpha}_j)$$
$$\forall j \in \{\text{fix, sac, sp}\}. \quad (5)$$

Table 1. Summary of model parameters.

State transition $\boldsymbol{\alpha}_{st}$	State duration $\boldsymbol{\alpha}_{sd}$	Fixation $\boldsymbol{\alpha}_{fix}$	Saccade $\boldsymbol{\alpha}_{sac}$	Smooth Pursuit $\boldsymbol{\alpha}_{sp}$	Measurement $\boldsymbol{\alpha}_{m}$
$\mathbf{A}$	$\mathbf{q}_i = [p_i(1), ..., p_i(D_i)]$ for i $\in$ {fix, sac, sp}	$\boldsymbol{\Sigma}_{fix}$	$\{\mu_\lambda, \sigma_\lambda, \mu_\theta, \sigma_\theta\}^{sac}$, $\{\mathbf{B}_1^i, \mathbf{B}_2^i\}_{i=3}^{D_{sac}}, \boldsymbol{\Sigma}_{sac}$	$\{\mu_\lambda, \sigma_\lambda, \mu_\theta, \sigma_\theta\}^{sp}$, $\{\mathbf{C}_1^i, \mathbf{C}_2^i\}_{i=3}^{D_{sp}}, \boldsymbol{\Sigma}_{sp}$	$\mathbf{D}, \boldsymbol{\Sigma}_n$

3.2. Static Eye Gaze Estimation

Figure 5. Architecture of static gaze estimation network.

The raw gaze measurements $\mathbf{x}_t$ is estimated with a standard deep convolutional neural network (Fig. 5) [44, 45]. The input are left and right eyes (both of size 36×60) and the 6-dimension head pose information (rotation and translation: pitch, yaw, roll angles and x, y, z). The left and right eye branch share the same weights of the convolutional layers. Each convolution layer is followed by a max-pooling layer with size 2. RELU is used for the activation of fully-connected layers. Detailed layer configuration are as follows: CONV-R1, CONV-L1: 5×5/50, CONV-R2, CONV-L2: 5×5/100, FC-RT1: 512, FC-E1, FC-RT2: 256, FC-1: 500, FC-2: 300, FC-3: 100. For simplicity, we denote static gaze estimation as $\mathbf{x}_t = f(\mathbf{I}_t; \mathbf{w})$, where $\mathbf{I}$ and $\mathbf{w}$ are input frame and model parameters respectively.

3.3. Online Eye Gaze Tracking

Traditional static-based methods only output the measured gaze $\mathbf{x}$ from static gaze estimation network. In this work, we propose to output the true gaze $\mathbf{y}$ with the help of DGTN:

$$\begin{aligned}\mathbf{y}_t &= \arg\max p(\mathbf{y}_t|\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t) \\ &= \arg\max \int_{s_t} p(\mathbf{y}_t, s_t|\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t) ds_t \end{aligned} \quad (6)$$

Solving the problem in Eq. (6) directly is intractable because of the integral over the hidden state. Alternatively we propose to first draw samples of possible state s_t ([43]) from its posterior. Given state, gaze estimation is a standard inference problem of LDS or Kalman filter ([46]). The algorithm is summarized in Alg. 1.

3.4. Model Refinement

The static gaze estimation network is learned from subjects during the offline stage. They may not generalize well to new subjects or environments. Therefore we propose to leverage on the refined true gaze to refine the static gaze estimation network (last two fully-connected layers). The algorithm is illustrated in Alg. 2. Notice we do not use the exact values of $\mathbf{y}$, but instead assuming the temporal gaze distribution from the static network ($p(\mathbf{x}_t)$) matches the true gaze distribution ($p(\mathbf{y}_t)$). Similar to Fig. 3 (b) and (c), we treat the $x-t$ curve and $y-t$ curve as two categorical distributions($\mathbf{p} = [p_1, ..., p_T]$), whose range is from 1 to T, and the value p_i equals to the normalized gaze positions. By minimizing the KL-divergence between the two gaze distributions, we can gradually refine the parameters of the static network. The proposed algorithm may not give good accuracy in the beginning, but it can be performed incrementally and gives better predictions as we collect more frames.

Algorithm 1: Online eye tracking

while *getting a new frame* $\mathbf{I}_t$, **do**
- Draw samples of state s_t ([43]) from its posterior: $s_t^i \sim P(s_t|\mathbf{x}_{t-k}, ..., \mathbf{x}_t), \forall i = 1, ..., N$.
- According to the sample values of state s_t, using the corresponding LDS in Eq. (1)([46]) to predict the true gaze: $\mathbf{y}_t^i = \arg\max_{\mathbf{y}_t^i} P(\mathbf{y}_t^i|\mathbf{x}_{t-k}, ..., \mathbf{x}_t, s_t^i)\ \forall i = 1, ..., N$.
- Average the results from N samples: $\mathbf{y}_t \approx \frac{1}{N}\sum_{i=1}^N \mathbf{y}_t^i$.

Algorithm 2: Model refinement for static gaze estimation network.

1. Input: Static gaze estimation network $f(\cdot)$ with initial parameters $\mathbf{w}^0$.
2. **while** *getting a new frame* $\mathbf{I}_t$, **do**
   - Gather last k true gaze point $\mathbf{y}_t = (a_t, b_t)$ from Alg. 1 and construct two categorical distributions for horizontal and vertical gaze: $\mathbf{p}_x = \frac{1}{\sum a_i}[a_{t-k}, ..., a_t], \mathbf{p}_y = \frac{1}{\sum b_i}[b_{t-k}, ..., b_t]$.
   - Gather last k raw gaze point $(\hat{a}_t, \hat{b}_t) = f(\mathbf{I}_t; \mathbf{w})$ and construct bottom-up categorical distributions: $\mathbf{q}_x(\mathbf{w}) = \frac{1}{\sum \hat{a}_i}[\hat{a}_{t-k}, ..., \hat{a}_t]$, $\mathbf{q}_y(\mathbf{w}) = \frac{1}{\sum \hat{b}_i}[\hat{b}_{t-k}, ..., \hat{b}_t]$.
   - Update static gaze estimation network: $\mathbf{w}^t = \arg\min_{\mathbf{w}} D_{\mathcal{KL}}(\mathbf{p}_x||\mathbf{q}_x(\mathbf{w})) + D_{\mathcal{KL}}(\mathbf{p}_y||\mathbf{q}_y(\mathbf{w}))$, where $D_{\mathcal{KL}}(p||q) = \sum_i p(i) \log \frac{p(i)}{q(i)}$.

4. DynamicGaze Dataset

Existing datasets for gaze estimation and eye movement dynamics have little overlap. On one hand, gaze-related benchmark datasets are all frame-based. Subjects are asked to look at markers on the screen, where their face images and groundtruth gaze are recorded. However, there are no natural dynamic gaze patterns in the dataset. On the other hand, eye movement related datasets focus on collecting data while subjects watch natural video stimulus. Though the collected data involves dynamics, there are no bottom-up image measurements. To bridge the gap between these two fields, we construct a new dataset which records both images and groundtruth gaze positions while subjects perform natural operations (browsing websites, watching videos). Clear eye movement dynamics can be observed from the dataset.

To acquire the groundtruth gaze positions, we use a commercial eye tracker which runs at the back-end. In the meantime, the front-facing camera of the laptop records the video stream of the subjects. The video stream and the gaze stream are synchronized during post-processing. The Tobii 4C eye tracker gives less than 0.5 error after calibration, and we believe the accuracy is sufficient to construct a dataset for the webcam-based eye gaze tracking system.

4.1. Data collection procedure

We invite 15 male subjects and 5 female subjects, whose age ranges from 20 to 30, to participate in the dataset construction. We collected 3 sessions of data: 1) frame-based; 2) video-watching 3) website-browsing.

Frame-based. There are two purposes: 1) provide another benchmark for static eye gaze estimation and 2) train our generic static gaze estimation network. Subjects are asked to look at some random moving objects on the screen, the random moving objects are to ensure subjects' gaze spread on the entire screen. Each subject takes 3-6 trials at different days, locations. We also ask subjects to sit at different positions in front of the laptop to introduce more variations. Finally, we end up with around 370000 valid frames.

Video-watching. The subjects are asked to watch 10 video stimulus (Tab. 2) from 3 eye tracking research datasets. The collection procedure is similar to the previous session, and finally we collect a total of around 145000 valid frames.

Website-browsing. Similarly, subjects are asked to browse websites freely on the laptop for around $5-6$ minutes, and a total of around 130000 frames are collected.

4.2. Data visualization and statistics

Fig. 6 shows sample eye images from the 20 subjects. There are occlusions like glasses and reflections. Fig. 7 shows the spatial gaze distributions on a monitor with resolution 2880×1620. For frame-based data , the gaze appears uniformly distributed. For video-watching data, the gaze appears center-biased, which is the most common pattern when watching videos. Finally, for website-browsing, the gaze pattern is focused on the left side of the screen mainly due to the design of the website. Since the major goal of the dataset is to explore gaze dynamics, we also take a look at the dynamic gaze patterns from 8 subjects watching the same video stimuli. As shown in Fig. 8, different subjects share similar overall gaze patterns, though the exact values of horizontal and vertical gaze positions are different.

Table 2. Information about different video stimulus.

Dataset	Name	Description
CRCNS [47]	1. saccadetest	Dots moving across the screen.
	2. beverly07	People walking and running.
[48]	3. 01-car-pursuit	Car driving in a roundabout.
	4. 02-turning-car	Car turning around.
DIEM [49]	5. advert bbc4 bees	Flying bees on BBC logo.
	6. arctic bears	Arctic bears in the ocean.
	7. nightlife in mozambique	One crab hunting for fishes.
	8. pingpong no bodies	Pingpong bouncing around.
	9. sport barcelona extreme	Extreme sports cut.
	10. sport scramblers	Extreme sports for scramblers.

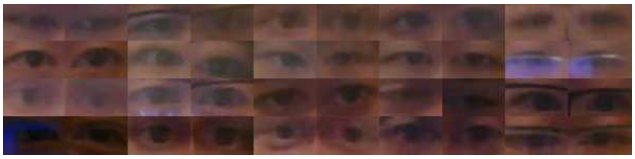
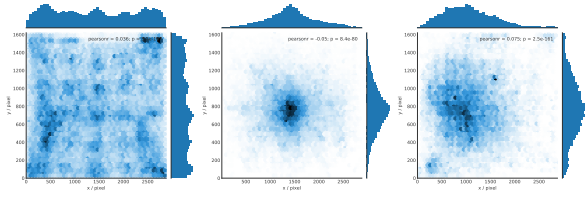


Figure 6. Sample eye images from the dataset.



(a) frame-based (b) video-watching (c) website-browsing

Figure 7. Spatial gaze distributions for the DynamicGaze dataset.

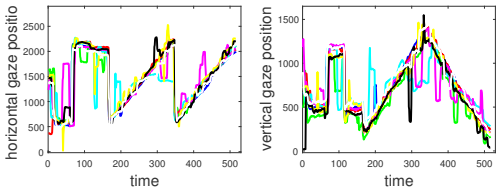


Figure 8. Sample dynamic gaze patterns from 8 subjects watching the same video.

5. Experiments and Analysis

For DGTN, the measurement model $P(\mathbf{x_t}|\mathbf{y_t})$ is learned with the data from DynamicGaze, where we have both groundtruth gaze $\mathbf{y}_t$ and measured gaze from the static gaze estimation network. The remaining part of the model is learned with the data from CRCNS [47], where we have the groundtruth state annotations s_t and the groundtruth gaze.

CRCNS consists of 50 video clips and 235 valid eye movement traces from 8 subjects. For the static gaze estimation network, we use Tensorflow as our backend engine.

Fixation is a one-order LDS, saccade, and smooth pursuit can be considered as second-order LDS, therefore the value k in Alg.1 is either 1 or 2. The value k in Alg.2 is set to 50 (around 2 seconds of data), where we use them to update the parameters of the static network. For overall gaze estimation, the static gaze estimation (GPU Tesla 54 K40c) takes less than 1 ms, while the online part (Alg. 1) with Intel Xeon CPU E5-2620 v3 @2.4GHz takes around 50-60 ms. In practice, for real-time processing, the model refinement runs with a separate thread other than the gaze estimation thread.

The performance is evaluated using the angular error in degree. We first compute the Euclidean pixel error on the monitor(2880×1620), which can be transformed to centimeter error err_d given monitor dimensions. The angular error is approximated by $err_a = \arctan(err_d/t_z)$, where t_z is the estimated depth of subject's head relative to the camera.

5.1. Baseline for Static Gaze Estimation Network

Table 3. Comparison of different input data channels.

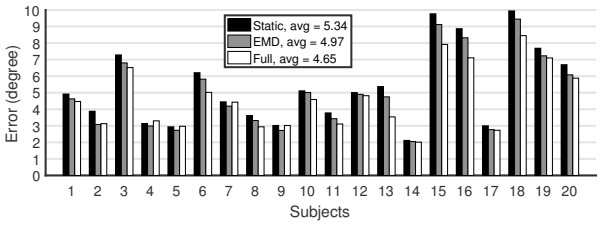
	L	R	F	L,R	L,R,F	L,R,P	L,R,F,P
Error	5.38	5.27	5.56	4.70	5.29	**4.27**	4.47

We experiment with different input combinations. As shown in Table 3, the symbol L, R, F, P represent left eye image, right eye image, face image, and head pose respectively. According to the results, we decide to use both eyes and head pose. To obtain head pose, we perform offline detection of the facial landmarks [50], then we can solve the head pose angle with a 3D shape model [51, 52]. Note that adding face is not helpful, since subjects have very different facial texture than eye texture, which makes it hard to generalize to new subjects. In addition, adding face may significantly increase the inference time.

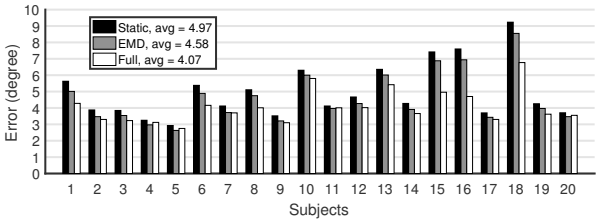
5.2. Evaluation on Different Model Components

The proposed model consists of two major components: 1) gaze estimation with eye movement dynamics and 2) refinement model to better fit current users/environments. To study the contributions of each component, we compare following 3 variants of the proposed model:

- Static: this model outputs the raw gaze prediction $\mathbf{x}$ and serves as the baseline.
- EMD (Eye movement dynamics): this model only uses eye movement dynamics (Alg. 1) without model refinement and output the true gaze prediction $\mathbf{y}$.
- Full: this is our full model contains both eye movement dynamics and model refinement.



(a) video-watching



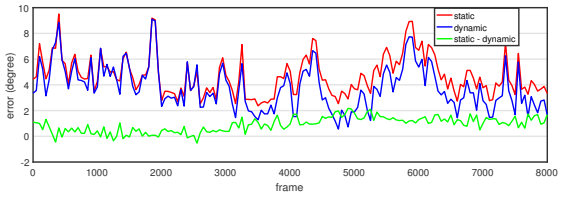
(b) website-browsing

Figure 9. Gaze estimation error for all subjects.

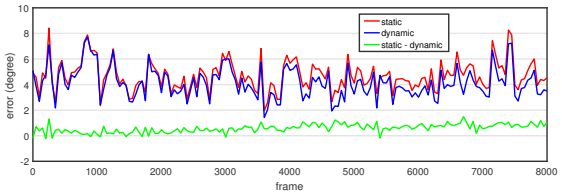
We perform cross-subject evaluation and Fig. 9 shows the performance of the 3 models. First, the Full model shows improved performance over the Static model for most subjects. The average estimation error reduces from 5.34 degrees to 4.65 degrees ((pitch, yaw) = (2.67, 3.81), 13% improvement) for video-watching and 4.97 degrees to 4.07 degrees ((pitch, yaw) = (2.23, 3.41), 18% improvement) for web-browsing. Second, compare EMD (gray bar) with Static (black bar), we can always achieve better results for both scenarios, demonstrating the importance of incorporating dynamics, especially in practical scenarios where user's gaze patterns have strong dynamics. The average improvements with eye movement dynamics are 6.9% and 7.9% for video-watching and website-browsing respectively. Third, the difference between Full (white bar) with EMD (gray bar) demonstrates the effect of Model Refinement. We can clearly observe that the Static model cannot generalize well to some subjects. With Model Refinement, we significantly reduce the error for some subjects (Eg. Subj 6, 15, 16, 18 in video-watching and Subj 15, 16, 18 in website-browsing). We also observe that model refinement may not always help, it may increase the error for some subjects (Eg. Subj 4, 5, 7 in video-watching). Averagely speaking, Model Refinement improves 6.4% and 11.2% for video-watching and website-browsing respectively. Overall, both components can help reduce the error of eye gaze estimation and combining the two further reduces the error.

5.3. Performance of gaze estimation over time

Fig. 10 shows the gaze estimation error over time. The error is averaged from all subjects from their first 8000 frames. For both scenarios, the improvement for the first period of time is small (sometimes even decrease), but gradually there is more significant improvements as we have more data.



(a) video-watching



(b) website-browsing

Figure 10. Gaze estimation error over time. Red curve represents error from Static model, green curve represents error from Full model and green curve shows the reduced error.

This demonstrates that with enough frames, the proposed method can significantly improve the accuracy of eye gaze estimation.

5.4. Comparison with different dynamic models

Table 4. Average error of all subjects with different dynamic models.

	Static	Mean	Median	LDS	s-LDS	RNN	Ours
Video	5.34	5.18	5.16	5.20	5.14	5.15	**4.97**
Web	4.97	4.85	4.84	4.70	4.66	4.71	**4.58**

In this experiment, we compare with several baseline dynamic models. The experimental results are illustrated in Table 4. First, we find incorporating dynamics outperforms the static method. Even the simple mean/median filters can improve the results. The LDS model trained on entire sequence without consideration of different eye movement types cannot give good results. Once we consider different eye movement types, the switching-LDS can improve the results even without duration modeling. RNN [53, 54] gives reasonably good results but ignores the characteristics of different eye movements and therefore our proposed method can still outperform it. Overall, we believe the proposed dynamic modeling can better explain the underlying eye movement dynamics and help improve the accuracy of eye gaze estimation.

5.5. Comparison with state-of-the-art

We compare with the state-of-the-art appearance-based method [27] for both within-dataset and cross-dataset experiments. Specifically, we re-implement the model in [27] using Tensorflow by following the same architecture and architecture-related hyperparameters. For training-related hyperparameters (e.g. learning rate, epochs), we do not follow the one in [27] and adjust them based on cross-validation.

Table 5. Comparison with state-of-the-art.

Exp.	Within-dataset		Cross-dataset	
	Video	Website	Video	Website
1. Static network (ours)	5.34	4.97	9.12	9.65
2. Static network ([27])	4.97	4.86	8.73	9.17
3. Static network (ours) + DGTN	4.65	4.07	7.15	7.87
4. Static network ([27]) + DGTN	4.51	4.00	7.05	7.59

For within-dataset experiments, the two models are trained on the frame-based data from DynamicGaze and are tested on web and video data from DynamicGaze. For cross-dataset experiments, the two models are trained with data from EyeDiap ([55]) and are tested on web and video data from DynamicGaze.

The results are shown in Table 5. We have following observations: 1) Compare Exp.1 and Exp.2, we can see both static networks give reasonable accuracy, and the more complex one ([27]) gives better performance than ours; 2) Compare Exp.2 and Exp.4, adding DGTN to static network significantly reduces the gaze estimation error; 3) similarly compare Exp.2 and Exp.4, adding DGTN module to state-of-the-art static network can still achieve better performance; 4) the improvement for cross-dataset setting is more significant than the within-dataset case, demonstrating better generalization by incorporating eye movement dynamics; 5) compare Exp.2 and Exp.3, we can find that our proposed method (Exp.3) outperforms current state-of-the-art (Exp.2), especially in the cross-dataset case.

6. Conclusion

In this paper, we propose to leverage on eye movement dynamics to improve eye gaze estimation. By analyzing the eye movement patterns when naturally interacting with the computer, we construct a dynamic gaze transition network that captures the underlying dynamics of fixation, saccade, smooth pursuit, as well as their durations and transitions. Combining top-down gaze transition prior from DGTN with the bottom-up gaze measurements from the deep model, we can significantly improve the eye tracking performance. Furthermore, the proposed method allows online model refinement which helps generalize to unseen subjects or new environments. Quantitative results demonstrate the effectiveness of the proposed method and the significance of incorporating eye movement dynamics into eye tracking.

Acknowledgments: The work described in this paper is supported in part by NSF award (IIS 1539012) and by RPI-IBM Cognitive Immersive Systems Laboratory (CISL), a center in IBM's AI Horizon Network.

References

[1] A. L. Yarbus, "Eye movements during perception of complex objects," in *Eye movements and vision*, pp. 171–211, Springer, 1967. 1

[2] W. A. W. Adnan, W. N. H. Hassan, N. Abdullah, and J. Taslim, "Eye tracking analysis of user behavior in online social networks," in *International Conference on Online Communities and Social Computing*, pp. 113–119, Springer, 2013. 1

[3] G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Online community detection in social sensing," in *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 617–626, ACM, 2013. 1

[4] J. Tang, X. Shu, G.-J. Qi, Z. Li, M. Wang, S. Yan, and R. Jain, "Tri-clustered tensor completion for social-aware image tag refinement," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1662–1674, 2017. 1

[5] G.-J. Qi, C. C. Aggarwal, and T. Huang, "Link prediction across networks by biased cross-network sampling," in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 793–804, IEEE, 2013. 1

[6] J. H. Goldberg, M. J. Stimson, M. Lewenstein, N. Scott, and A. M. Wichansky, "Eye tracking in web search tasks: design implications," in *Proceedings of the 2002 symposium on Eye tracking research & applications*, pp. 51–58, ACM, 2002. 1

[7] X. Wang, T. Zhang, G.-J. Qi, J. Tang, and J. Wang, "Supervised quantization for similarity search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2018–2026, 2016. 1

[8] S. Chang, G.-J. Qi, C. C. Aggarwal, J. Zhou, M. Wang, and T. S. Huang, "Factorized similarity learning in networks," in *2014 IEEE International Conference on Data Mining*, pp. 60–69, IEEE, 2014. 1

[9] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer vision and image understanding*, vol. 98, no. 1, pp. 4–24, 2005. 1

[10] K. Wang, R. Zhao, and Q. Ji, "Human computer interaction with head pose, eye gaze and body gestures," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 789–789, IEEE, 2018. 1

[11] R. Zhao, K. Wang, R. Divekar, R. Rouhani, H. Su, and Q. Ji, "An immersive system with multi-modal human-computer interaction," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 517–524, IEEE, 2018. 1

[12] R. R. Divekar, M. Peveler, R. Rouhani, R. Zhao, J. O. Kephart, D. Allen, K. Wang, Q. Ji, and H. Su, "Cira: An architecture for building configurable immersive smart-rooms," in *Proceedings of SAI Intelligent Systems Conference*, pp. 76–95, Springer, 2018. 1

[13] S. Fiedler and A. Glöckner, "The dynamics of decision making in risky choice: An eye-tracking analysis," *Frontiers in psychology*, vol. 3, p. 335, 2012. 1

[14] S. Martinez-Conde, J. Otero-Millan, and S. L. Macknik, "The impact of microsaccades on vision: towards a unified theory of saccadic function," *Nature Reviews Neuroscience*, vol. 14, no. 2, p. 83, 2013. 1

[15] D. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," 2010. 1

[16] K. Wang and Q. Ji, "3d gaze estimation without explicit personal calibration," *Pattern Recognition*, 2018. 1

[17] D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head," in *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, vol. 2, pp. II–451, IEEE, 2003. 1

[18] K. Wang, S. Wang, and Q. Ji, "Deep eye fixation map learning for calibration-free eye gaze tracking," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pp. 47–55, ACM, 2016. 1, 2

[19] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Transactions on biomedical engineering*, vol. 53, no. 6, pp. 1124–1133, 2006. 1

[20] K. Wang and Q. Ji, "Hybrid model and appearance based eye tracking with kinect," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pp. 331–332, ACM, 2016. 1

[21] X. Xiong, Q. Cai, Z. Liu, and Z. Zhang, "Eye gaze tracking using an rgbd camera: A comparison with a rgb solution," *UBICOMP*, 2014. 1

[22] K. Wang and Q. Ji, "Real time eye gaze tracking with 3d deformable eye-face model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1003–1011, 2017. 1

[23] L. Jianfeng and L. Shigang, "Eye-model-based gaze estimation by rgb-d camera," in *CVPR Workshops*, 2014. 1

[24] K. Wang and Q. Ji, "Real time eye gaze tracking with kinect," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 2752–2757, IEEE, 2016. 1

[25] K. Wang, R. Zhao, and Q. Ji, "A hierarchical generative model for eye image synthesis and eye gaze estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 440–448, 2018. 1

[26] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4511–4520, 2015. 1

[27] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2176–2184, 2016. 1, 2, 8

[28] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets," *Machine Vision and Applications*, vol. 28, no. 5-6, pp. 445–461, 2017. 1

[29] T. Fischer, H. Jin Chang, and Y. Demiris, "Rt-gene: Real-time eye gaze estimation in natural environments," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 334–352, 2018. 1

[30] Y. Cheng, F. Lu, and X. Zhang, "Appearance-based gaze estimation via evaluation-guided asymmetric regression," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 100–115, 2018. 1

[31] K. Wang, R. Zhao, H. Su, and Q. Ji, "Generalizing eye tracking with bayesian adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[32] Y. Sugano, Y. Matsushita, and Y. Sato, "Appearance-based gaze estimation using visual saliency," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 329–341, 2013. 2

[33] J. Chen and Q. Ji, "A probabilistic approach to online eye gaze tracking without explicit personal calibration," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1076–1086, 2015. 2

[34] M. Cerf, J. Harel, W. Einhäuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Advances in neural information processing systems*, pp. 241–248, 2008. 2

[35] Q. Zhao, S. Chang, F. M. Harper, and J. A. Konstan, "Gaze prediction for recommender systems," in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 131–138, ACM, 2016. 2

[36] Y. Feng, G. Cheung, W.-t. Tan, and Y. Ji, "Hidden markov model for eye gaze prediction in networked video streaming," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pp. 1–6, IEEE, 2011. 2

[37] A. T. Duchowski, "Eye tracking methodology," *Theory and practice*, vol. 328, 2007. 2

[38] M. Nyström and K. Holmqvist, "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data," 2010. 2

[39] E. Tafaj, G. Kasneci, W. Rosenstiel, and M. Bogdan, "Bayesian online clustering of eye movement data," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 285–288, ACM, 2012. 2

[40] L. Larsson, M. Nyström, R. Andersson, and M. Stridh, "Detection of fixations and smooth pursuit movements in high-speed eye-tracking data," *Biomedical Signal Processing and Control*, vol. 18, pp. 145–152, 2015. 2

[41] T. Santini, W. Fuhl, T. Kübler, and E. Kasneci, "Bayesian identification of fixations, saccades, and smooth pursuits," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pp. 163–170, ACM, 2016. 2

[42] O. Le Meur and Z. Liu, "Saccadic model of eye movements for free-viewing condition," *Vision research*, vol. 116, pp. 152–164, 2015. 3

[43] K. P. Murphy, "Hidden semi-markov models (hsmms)," 2002. 4, 5

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012. 5

[45] T. Zhang, G.-J. Qi, B. Xiao, and J. Wang, "Interleaved group convolutions," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4373–4382, 2017. 5

[46] K. P. Murphy and S. Russell, "Dynamic bayesian networks: representation, inference and learning," 2002. 5

[47] R. Carmi and L. Itti, "The role of memory in guiding attention during natural vision," *Journal of vision*, vol. 6, no. 9, pp. 4–4, 2006. 6

[48] K. Kurzhals, C. F. Bopp, J. Bässler, F. Ebinger, and D. Weiskopf, "Benchmark data for evaluating visualization and analysis techniques for eye tracking for video stimuli," in *Proceedings of the fifth workshop on beyond time and errors: novel evaluation methods for visualization*, pp. 54–60, ACM, 2014. 6

[49] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson, "Clustering of gaze during dynamic scene viewing is predicted by motion," *Cognitive Computation*, vol. 3, no. 1, pp. 5–24, 2011. 6

[50] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)," in *International Conference on Computer Vision*, vol. 1, p. 4, 2017. 7

[51] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 607–626, 2009. 7

[52] K. Wang, Y. Wu, and Q. Ji, "Head pose estimation on low-quality images," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 540–547, IEEE, 2018. 7

[53] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010. 8

[54] H. Hu and G.-J. Qi, "State-frequency memory recurrent neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1568–1577, JMLR. org, 2017. 8

[55] K. A. F. Mora, F. Monay, and J.-M. Odobez, "Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 255–258, ACM, 2014. 8