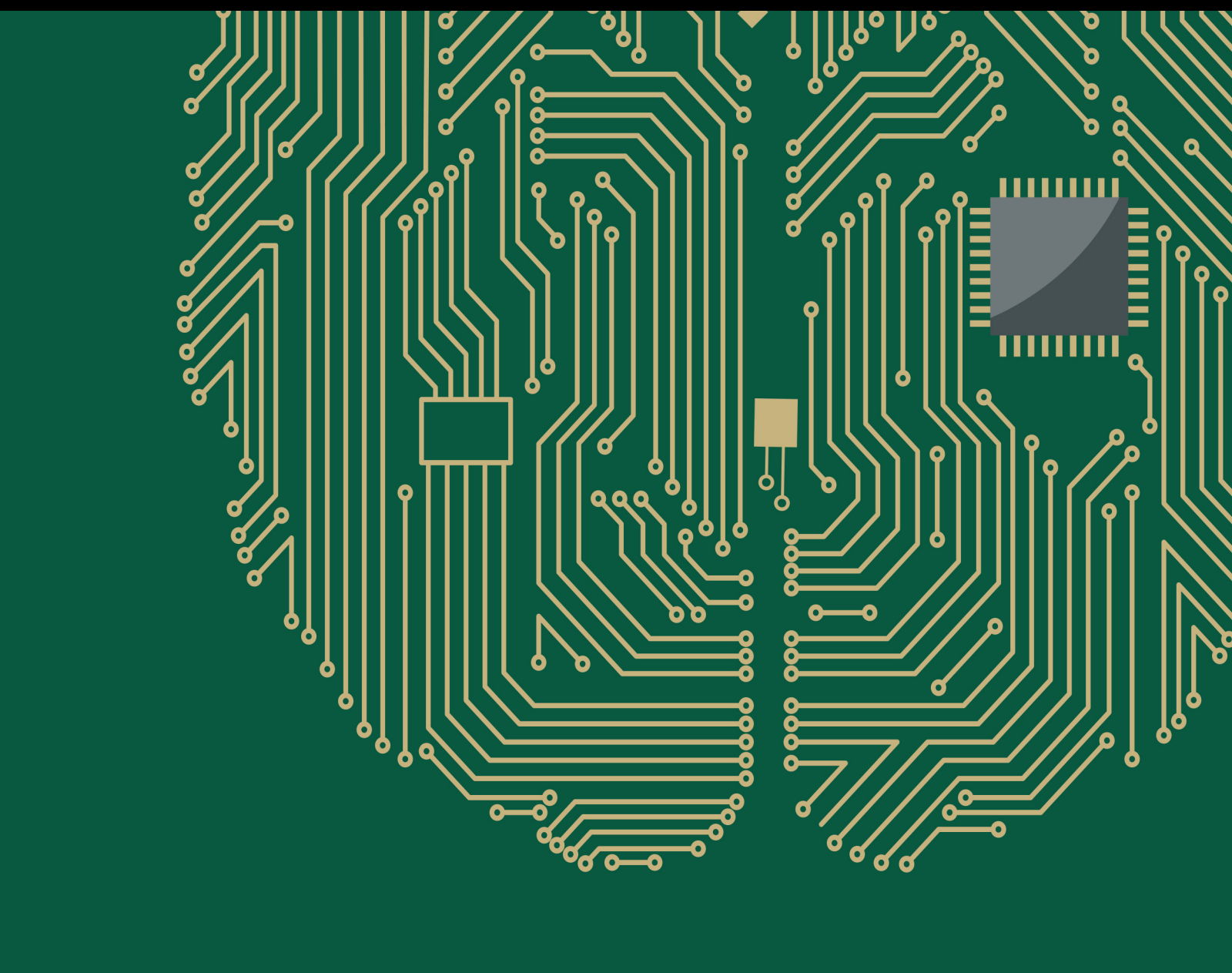
Computational Intelligence and Neuroscience

Advances in Eye Tracking Technology: Theory, Algorithms, and Applications

Guest Editors: Hong Fu, Ying Wei, Francesco Camastra, Pietro Arico, and Hong Sheng



Advances in Eye Tracking Technology: Theory, Algorithms, and Applications

Computational Intelligence and Neuroscience

Advances in Eye Tracking Technology: Theory, Algorithms, and Applications

Guest Editors: Hong Fu, Ying Wei, Francesco Camastra, Pietro Arico, and Hong Sheng

Editorial Board

Contents

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 7831469, 2 pages
http://dx.doi.org/10.1155/2016/7831469

Editorial

Advances in Eye Tracking Technology: Theory, Algorithms, and Applications

Hong Fu,[1] Ying Wei,[2] Francesco Camastra,[3] Pietro Arico,[4] and Hong Sheng[5]

[1]*Chu Hai College of Higher Education, New Territories, Hong Kong*
[2]*Shandong University, Jinan 250100, China*
[3]*Department of Science and Technology, University of Naples Parthenope, Centro Direzionale, Isola C4, 80134 Naples, Italy*
[4]*University of Rome "Sapienza", Piazzale Aldo Moro 5, 00185 Rome, Italy*
[5]*Missouri University of Science and Technology, Rolla, MO, USA*

Correspondence should be addressed to Hong Fu; hongfu@chuhai.edu.hk

Received 21 July 2016; Accepted 21 July 2016

The goal of eye tracking is to detect and measure the point of gaze (where one is looking) or the motion of eye(s) relative to the head. The eye tracking data obtained by an eye tracker provide new opportunities and potentials in a broad range of applications including human computer interaction, computer simulation/virtual reality, neuroscience, medical, and cognitive-behavioral research. In recent years, eye tracking technology has been undergoing rapid development with improvements in the accuracy, stability, and sampling rates. A number of technologies and techniques are now available, including head-mounted, glass, table-mounted, and embedded systems, and with these advances new opportunities and applications are emerging. This special issue aims to bring together theoretical and practical perspectives in the area of eye tracking technology to present and discuss the latest technological developments and to inspire interaction and creation.

The issue received fifteen submissions; each qualified submission was reviewed by two international reviewers that we warmly thank for their time and efforts. Seven papers have been accepted for the publication.

"Learning to Model Task-Oriented Attention" by X. Zou et al. describes a model of saliency based on bottom-up image features and target position feature. Experimental results demonstrate the importance of the target information in the prediction of task-oriented visual attention.

"Characterization of Visual Scanning Patterns in Air Traffic Control" by S. N. McClung and Z. Kang defines new concepts to systematically filter complex visual scanpaths into simpler and more manageable forms and develops procedures to map visual scanpaths with linguistic inputs to reduce the human judgement bias during interrater agreement. The developed concepts and procedures were applied to investigating the visual scanpaths of expert ATCs using scenarios with different aircraft congestion levels. The findings show that the scanpaths filtered at the highest intensity led to more consistent mapping with the ATCs' linguistic inputs, the pattern classification occurrences differed between scenarios, and increasing aircraft congestion caused increased scan times and aircraft pairwise comparisons. These results provide a foundation for better characterizing complex scanpaths in a dynamic task and automating the analysis process.

"EyeTribe Tracker Data Accuracy Evaluation and Its Interconnection with Hypothesis Software for Cartographic Purposes" by S. Popelka et al. introduced a possible combination of Hypothesis software with EyeTribe tracker. A new software platform was presented which connects eye tracking device with an experiment builder. Experimental results showed that the mixed research design combines the advantages of quantitative and qualitative methods.

"Low Cost Eye Tracking: The Current Panorama" by O. Ferhat and F. Vilariño provided an overview of remote visible

light gaze trackers and challenges in this area. The authors also analyzed the explored techniques from various perspectives such as calibration strategies, head pose invariance, and gaze estimation techniques.

"Learning-Based Visual Saliency Model for Detecting Diabetic Macular Edema in Retinal Image" by X. Zou et al. presents a learning-based visual saliency model method for detecting diagnostic diabetic macular edema regions of interest in retinal image. The method introduces the cognitive process of visual selection of relevant regions that arises during an ophthalmologist's image examination. The proposed method outperforms state-of-the-art saliency models and salient region detection approaches derived for natural images.

"Real-Time Control of a Video Game Using Eye Movements and Two Temporal EEG Sensors" by A. N. Belkacem et al. presents an algorithm able to classify six classes of eye movement by using only two temporal EEG electrodes, thus, in a noninvasive way. Moreover, this algorithm has been tested on real-time applications, in particular the control, by means of the eye movements, of a screen cursor and then of a character in a video game. Results showed that the proposed algorithm had an efficient response speed demonstrating its efficacy and robustness in real-time control.

"Designs and Algorithms to Map Eye Tracking Data with Dynamic Multielement Moving Objects" by Z. Kang et al. presents algorithms to address the eye tracking analysis issues when participants interrogate dynamic multielement objects and when eye trackers are incapable of providing exact eye fixation coordinates. The approach was tested in air traffic control (ATC) operations and the more accurate results could be obtained for eye tracking data analysis.

We hope that the readers of this journal will find in the issue interesting papers and that this can encourage and foster further research on eye tracking technology.

Hong Fu
Ying Wei
Francesco Camastra
Pietro Aricò
Hong Sheng

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 9354760, 18 pages
http://dx.doi.org/10.1155/2016/9354760

Research Article

Designs and Algorithms to Map Eye Tracking Data with Dynamic Multielement Moving Objects

Ziho Kang,[1] Saptarshi Mandal,[1] Jerry Crutchfield,[2] Angel Millan,[2] and Sarah N. McClung[3]

[1]*School of Industrial and Systems Engineering, University of Oklahoma, 202 West Boyd Street, Norman, OK 73019, USA*
[2]*Aerospace Human Factors Research Division, Civil Aerospace Medical Institute AAM-520, Federal Aviation Administration, P.O. Box 25082, Oklahoma City, OK 73125, USA*
[3]*School of Electrical and Computer Engineering, University of Oklahoma, 110 W. Boyd Street, Devon Energy Hall 150, Norman, OK 73019-1102, USA*

Correspondence should be addressed to Ziho Kang; zihokang@ou.edu

Received 28 November 2015; Revised 15 March 2016; Accepted 16 May 2016

Academic Editor: Hong Fu

Design concepts and algorithms were developed to address the eye tracking analysis issues that arise when (1) participants interrogate dynamic multielement objects that can overlap on the display and (2) visual angle error of the eye trackers is incapable of providing exact eye fixation coordinates. These issues were addressed by (1) developing dynamic areas of interests (AOIs) in the form of either convex or rectangular shapes to represent the moving and shape-changing multielement objects, (2) introducing the concept of AOI gap tolerance (AGT) that controls the size of the AOIs to address the overlapping and visual angle error issues, and (3) finding a near optimal AGT value. The approach was tested in the context of air traffic control (ATC) operations where air traffic controller specialists (ATCSs) interrogated multiple moving aircraft on a radar display to detect and control the aircraft for the purpose of maintaining safe and expeditious air transportation. In addition, we show how eye tracking analysis results can differ based on how we define dynamic AOIs to determine eye fixations on moving objects. The results serve as a framework to more accurately analyze eye tracking data and to better support the analysis of human performance.

1. Introduction

Eye tracking research is useful for evaluating usability or analyzing human performance and more importantly understanding underlying cognitive processes based on the eye-mind hypothesis [1]. This hypothesis asserts that what we observe when performing a task is highly correlated with our cognitive processes. Thus, eye tracking research has been conducted in diverse fields to investigate how objects or spatially fixed areas are interrogated [2–7]. For example, an air traffic control specialist (ATCS) must timely detect and control multiple aircraft on a radar display in order to maintain a safe and expeditious flow of air traffic. Through eye tracking data, we can identify which aircraft the ATCS interrogates and what visual search pattern the ATCS applies.

However, the analysis of eye tracking data for a task that requires interrogating moving objects (e.g., an ATCS controlling multiple moving aircraft on a radar display or a weather forecaster determining whether to issue a warning by observing the weather features on a radar display) can be difficult due to the different characteristics of the moving objects and the limited capabilities of the eye tracking system. Furthermore, eye tracking analysis becomes more difficult if the object's overall shape can change due to the shape change of the object's elements or the physical relocation of its elements (e.g., an aircraft on a radar screen is composed of elements such as a vector line and a data block, and the length of the vector line can change due to the aircraft speed change, or the data block can be repositioned by the ATCS). The details of the issues are as follows.

In order to map and analyze the eye tracking data for such a task described above, different characteristics of those moving objects need to be identified (Figure 1). Objects can have irregular shapes and sizes and different movement

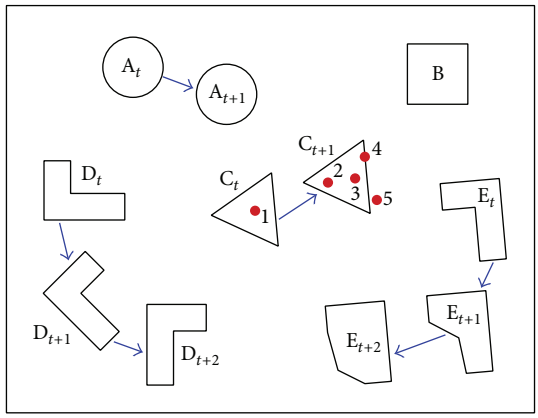


Figure 1: Characteristics of multiple moving objects: each object is in motion except for object "B." "A_t" indicates a circular object at time t and "A_{t+1}" indicates the change of its location at time $t + 1$. "D" is an object rotating clockwise, and "E" is an object changing its shape. The red dots on and around "C" indicate the order of eye fixations at times t (eye fixation 1) and $t + 1$ (eye fixations 2 to 5).

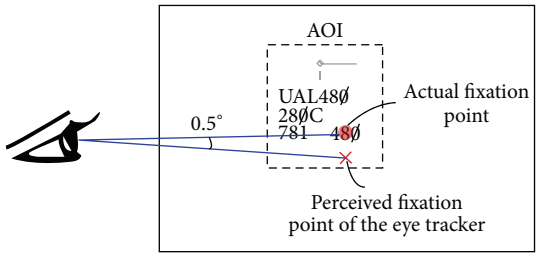


Figure 2: Area of interest (AOI) and visual angle accuracy error: The AOI approximates the shape of the object and should be slightly bigger than the original object size considering the visual angle error. The object consists of the aircraft itself (shown as a small diamond shape), the direction indicator (currently flying east), the data block (aircraft ID: UAL 480, altitude: cruising at 28000 ft., computer ID: 781, and speed: 480 knots), and the leader line which points to its corresponding aircraft.

characteristics and can be at close proximity or overlap with one another as time progresses. When the eye fixation data is collected, we can overlay the data with the objects to determine whether an eye fixation occurred on the object.

Eye tracking systems return pixel-based coordinates where the eyes fixated; however, we are more interested in (1) whether eye fixations occurred on the objects of interests as well as (2) the order of the eye fixations among those objects of interest. Specifically, we need to consider the following issues when mapping the pixel-based eye fixations with the multielement objects on a display.

One of the difficulties with mapping the eye tracking data to the objects is due to the visual angle accuracy of the eye trackers (Figure 2). A visual angle accuracy (expressed in degrees) is defined as the deviation of coordinates, collected from the eye tracker, from the actual location on which the individual fixated [8, 9] (e.g., 0.4~0.5° [10–12]) when using displays that are approximately below 16 (horizontal length) × 12 (vertical length) inches (or 22 inches diagonally) in size. For example, if a display is observed from 1 meter away with visual angle accuracy of 0.5°, then we can have up to 1 cm of error where the eyes fixated on. Therefore, observing the eye fixations shown as red dots in Figure 1, in addition to the first four eye fixations, we could also determine that the fifth eye fixation may have occurred on object "C." In addition to the inherent error of eye tracking systems, accuracy error can also be affected by experimental conditions.

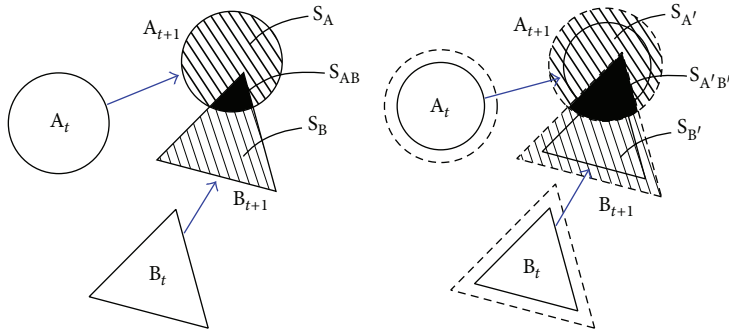
For example, in the actual air traffic control rooms, ATCSs sit close to a large monitor (i.e., 19.83 × 19.83 inches) in order to better detect and control multiple (i.e., sometimes up to 50 or more) aircraft within their sector. For such an environment, the accuracy of the eye tracker can drastically decrease. These issues occur when measuring eye tracking data not only in an air traffic control task, but also in other various tasks such as during driving or during a virtual simulation of offshore oil and gas operations. Therefore, the visual angle accuracy is not fixed at 0.5° and can vary based on the experimental conditions when we pursue high face validity.

In addition, the mapping of eye tracking data to moving objects can be difficult if there are multiple small objects moving on the display and each object is composed of several elements (e.g., the aircraft position symbol (or target), vector line, and data block). To accommodate the complex shapes of objects as well as the visual angle accuracy, the concept of an area of interest (AOI) can be applied. An AOI is a convex shape that can approximate and represent the complex object shape and can be simple shapes such as circles and rectangles. For example, the AOI can be fixed rectangular areas [5, 6, 13] or moving rectangular areas [9, 14] on a display based on the task types. Note that the size of an AOI should be slightly enlarged to consider the visual angle accuracy [9, 14].

To determine whether an eye fixation occurred on an object, we need to consider two aspects. First, the eye fixation should have occurred within the visual angle error range (e.g., 0.5° from all edge points of an object). Second, there should be no other object or background image to which the eye fixation occurred. In other words, if two objects are in close proximity, it can be difficult to determine which object the participant was interrogating. Even if the objects arrived from different locations, they can come into close proximity and even overlap as time progresses (Figure 3). Although considerable research was conducted to investigate the eye movements of air traffic control operations [15–18], it was limited to creating spatially fixed AOIs or did not elaborate on how overlapping issues were addressed.

Additionally, the mapping issue becomes more complex if the shapes of the multielement objects change. For example, if two aircraft are approaching close proximity, the aircraft position symbols (or targets) as well as the data blocks can overlap, and then an ATCS can reposition the data block (Figure 4). The data block can be repositioned in eight directions relative to the aircraft position symbol (e.g., from the bottom of the target itself to the top or right upper corner of the target) as well as increased in distance (e.g., from 0.5 cm away to 5 cm away).

In this paper, we present designs and algorithms to address the issues raised to facilitate the analysis of the eye



(a) Overlapping objects over time (b) Overlapping AOIs over time

FIGURE 3: Overlapping objects and defined AOIs over time: the AOIs are designed slightly larger than the objects themselves to accommodate the visual angle error. The overlapping areas are denoted as S_{AB} and $S_{A'B'}$.

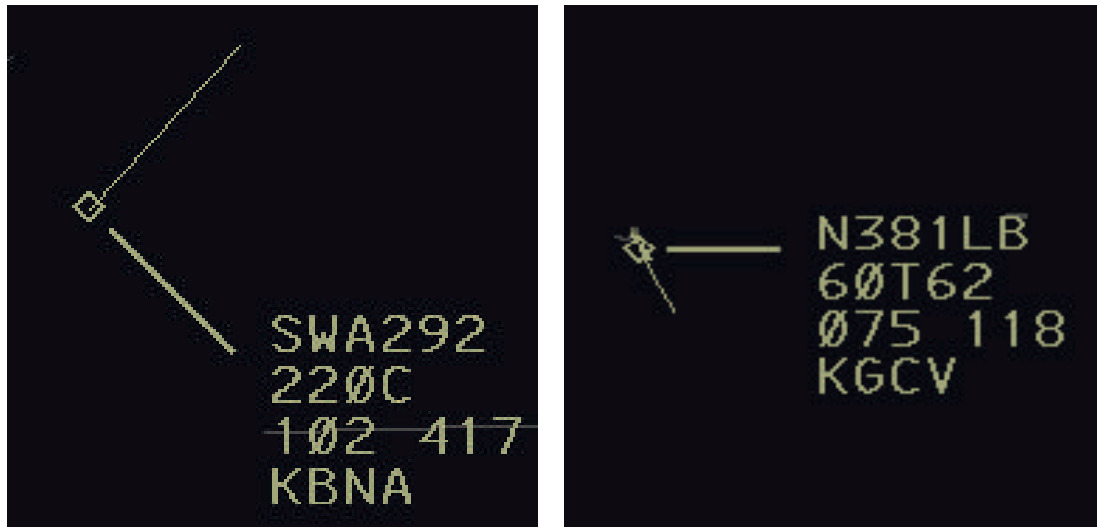


FIGURE 4: The overall shape change of an aircraft position indicator along with the data block. An ATCS can freely move around the data block since the leader line connects the data block to its aircraft. The overall shape also changes if the aircraft changes its direction.

tracking data for tasks that involve interrogating multielement moving objects that can change their overall shape and overlap with one another by considering different shapes and sizes of the AOIs that are fitted to represent the objects.

2. Conceptual Designs and Algorithms

The main features of our approach are to (1) develop dynamic AOIs that continuously fit the multielement objects into convex or rectangular shapes whenever the objects' overall shapes or locations change, (2) modify the size of the AOIs (through the concept of AOI gap tolerance) to consider the visual angle error, (3) map the pixel-coordinate based eye fixations with the AOIs, and (4) define eye fixations on overlapping AOIs. Specific to air traffic control operations, the designs and algorithms create AOIs based on matching the pixel-coordinates from the flight data block, target, and vector lines with the pixel-coordinates of the eye fixations. Figure 5 represents the data processing flowchart of the overall methodology. The flowchart consists of seven major steps, which are discussed in detail in the subsequent sections. Note that the introduced algorithm is based on discretized movements of the moving objects, and the background (scene) is fixed.

Step 1. Collect and preprocess simulation and eye tracking data.

Step 1.1 (collect and preprocess simulation data). Assume the simulation scenario is of m duration in minutes, given an update rate (UR) in seconds (e.g., 1 second), defined as the refresh rate of the objects' locations and shapes on a display; the total duration of a scenario can be divided into UR$\times m\times$60 time frames in seconds. Thus, if we want to represent the m minutes scenario into discrete time frames we can represent it as

$$T = \{\mathrm{UR}, \mathrm{UR}\times 2, \mathrm{UR}\times 3, \ldots, \mathrm{UR}\times m\times 60\}, \quad (1)$$

where T represents the time frame counter in seconds.

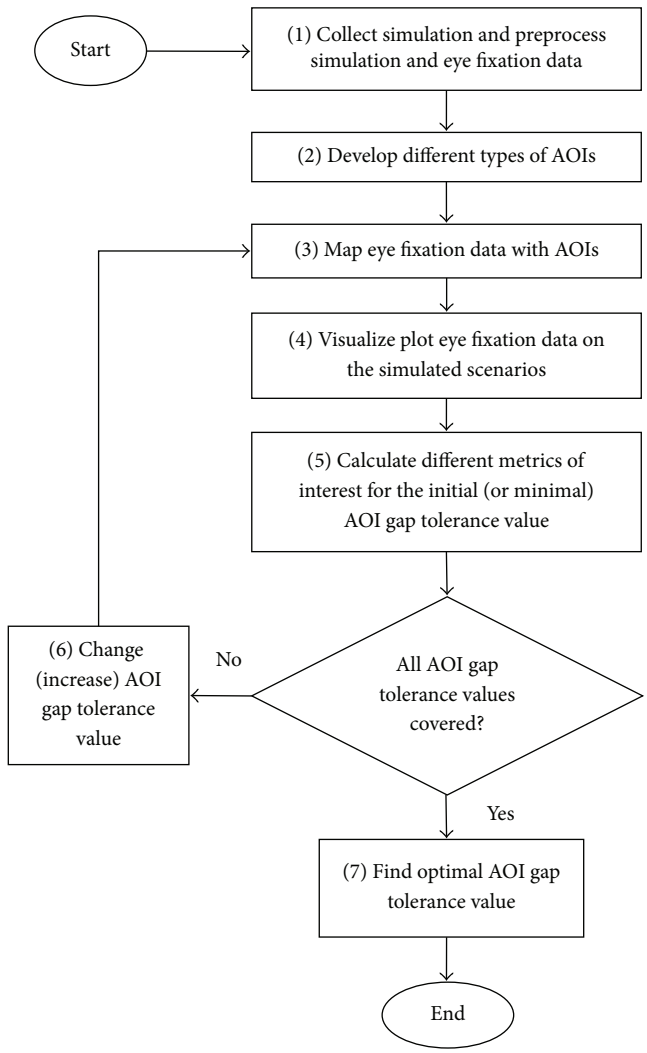


Figure 5: Data processing flowchart.

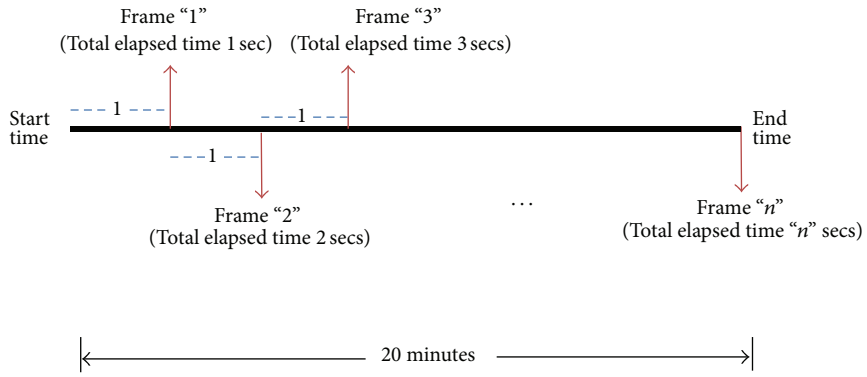


Figure 6: Discretization of the simulation video into time frames.

Figure 6 represents an example of the discretization process of the simulation output for a 20-minute duration. Note an observable (or systematic) discrete movement of the object (e.g., aircraft or radar display). In other words, no change in position occurs within a time frame; for example, suppose the simulation starts at 0 seconds, the next change in positon of the aircraft will occur at the end of the first second, and the next change will be at the end of two seconds

Table 1: Example of eye fixation data.

X pos (pixels)	Y pos (pixels)	Start time (secs)	Stop time (secs)	Duration (secs)
1047	1668	0.35	0.466	0.116
628	1255	0.816	1.15	0.334
852	1174	1.5	2.233	0.733
1150	1690	2.666	2.916	0.25
1162	1721	2.933	3.416	0.483

and so on. After discretizing the time frames as part of the simulation data preprocessing step, the corresponding multielement object data are identified for each time frame. Let P be the set that contains all the information of the multielements for the total time duration. Then P can be represented as

$$P_{N_T,T} = \left\{p_{n_{\mathrm{UR}},\mathrm{UR}}, p_{n_{\mathrm{UR}},\mathrm{UR}\times 2}, \ldots, p_{n_{\mathrm{UR}\times m\times 60},\mathrm{UR}\times m\times 60}\right\}, \quad (2)$$

where $P_{N_T,T}$ is the set of multielement objects present for each time frame.

Step 1.2 (collect and preprocess eye fixation data). The eye fixation data needs to be processed according to the time discretization strategy used for processing the simulation data. Table 1 represents a small sample of eye fixation data. The first and second columns represent the horizontal and vertical pixel-coordinates of the eye fixations, respectively. The third and fourth columns show the start and stop time of an eye fixation. The fifth column represents the time duration of an eye fixation. The start and stop time values can be used to determine the time frame in which the eye fixations occurred.

The eye fixations during a time frame can be described as

$$E_{M_T,T} = \left\{e_{m_1,\mathrm{UR}}, e_{m_2,\mathrm{UR}\times 2}, \ldots, e_{m_{\mathrm{UR}\times m\times 60},\mathrm{UR}\times m\times 60}\right\}, \quad (3)$$

where $E_{M_T,T}$ is the set of eye fixations that occurred for each time frame.

Figure 7 shows an example of eye fixation durations that occurred over the time frames. The time frames are based on the object movement update rate (i.e., objects would make discrete short burst of movements), and eye fixation durations can either fall within a time frame or stretch over more than one time frame.

Step 2 (develop different types of AOIs). Based on the preprocessed data from Step 1, different types of AOIs were developed. Two types of dynamic AOIs are considered: convex AOI and rectangular AOI. The rectangular AOI is an adaptation from [9], and in this research the shape and size of the rectangular AOI change based on each time frame. The convex AOI was developed by calculating the convex hull [19, 20] of the set of coordinate points used to represent each multielement object. The convex AOIs change their shapes and sizes based on each time frame as well. Figure 8 represents the two different types of AOIs (convex and rectangular) for a multielement object. Thus, if an eye fixation occurs within a dynamic AOI, then we conclude that an eye fixation occurred on the multielement moving object.

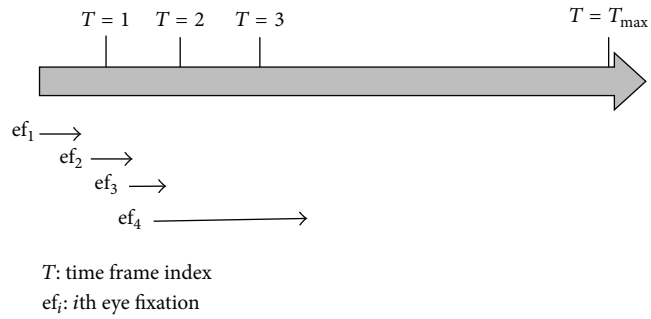


Figure 7: Example of eye fixation durations: the lengths of the arrows represent the eye fixation durations.

To define a parameter that governs the size of the buffer, we define the buffer as the "AOI gap tolerance (AGT)." Since any given AOI corresponds to only one multielement object, $P_{N_T,T}$ can be substituted by $\mathrm{AOI}_{N_T,T}$, the set of AOIs during a time frame, as

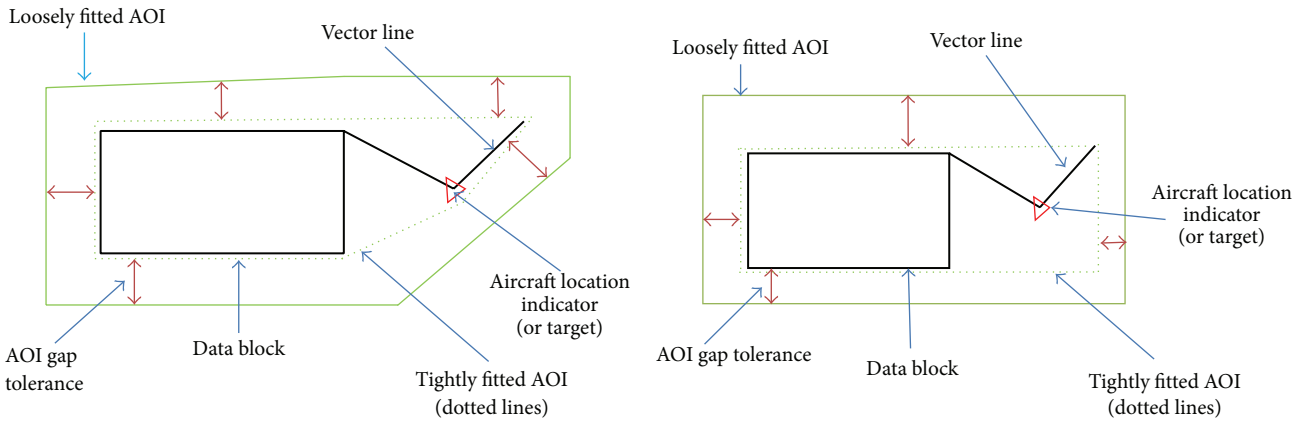
$$\mathrm{AOI}_{N_T,T} = \left\{\mathrm{aoi}_{n_{\mathrm{UR}},\mathrm{UR}}, \mathrm{aoi}_{n_{UR\times 2},\mathrm{UR}\times 2}, \ldots, \mathrm{aoi}_{n_{\mathrm{UR}\times m\times 60},\mathrm{UR}\times m\times 60}\right\}. \quad (4)$$

Step 3 (map eye fixation data with AOIs). The "AOI mapping (AM)" performs a match between the eye fixation set and the AOI set during the same time frame. AOI mapping identifies whether the eye fixations fell within the boundaries of the AOIs by comparing the coordinates. The AM can be expressed as

$$\mathrm{AM} : E_{M_T,T} \longrightarrow \mathrm{AOI}_{N_T,T}. \quad (5)$$

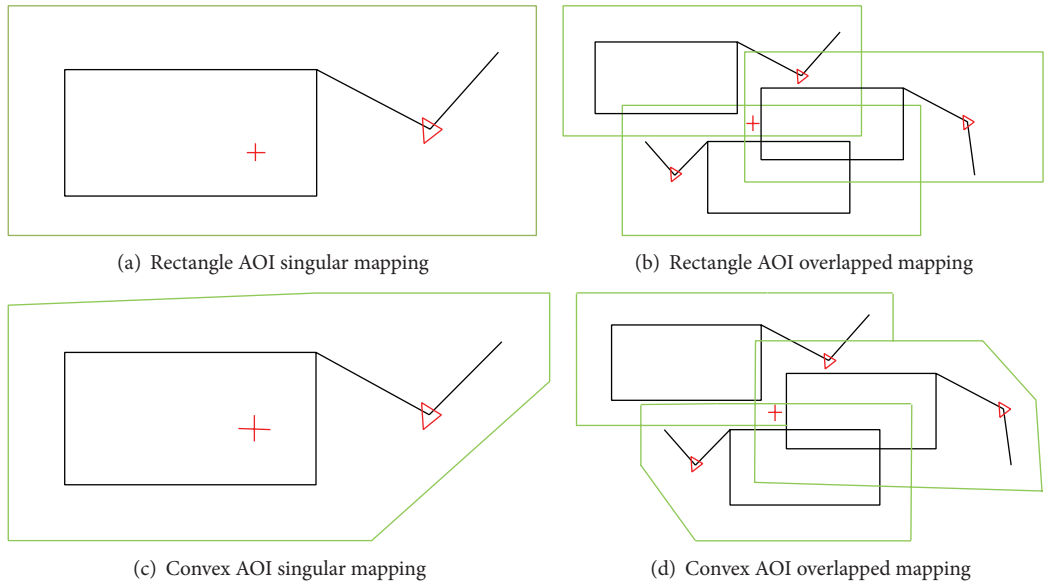
The functional mapping described in (5) is called a many-to-many mapping. Many-to-many mapping refers to the fact that eye fixations can be mapped to more than one AOI index and similarly AOIs can also be mapped to more than one eye fixation during a time frame. For example, in a single time frame, two or more eye fixations (that have different pixel-coordinates) can occur within a single AOI, or two or more AOIs can share a single eye fixation (when overlapping). The resulting mapped AOIs during the time frame t can be expressed as $\mathrm{AM}(e_{m_t,t}) = \mathrm{aoi}_{n_t,t}$. The collection of all mapped AOIs can be defined as a "mapped AOI set (MA)" and be written as

$$\mathrm{MA}_{I,T} = \left\{\mathrm{ma}_{i,t} \mid \mathrm{AM}\left(e_{m_t,t}\right) = \mathrm{aoi}_{n_t,t}, \mathrm{ma}_{i,t} \in \mathrm{aoi}_{n_t,t}\right\}, \quad (6)$$



(a) Convex AOI (b) Rectangular AOI

FIGURE 8: Example of a multielement object (black and red) represented using dynamic AOIs (green solid and dotted lines surrounding the object): the shape created when using dotted green lines represents a tightly fitted AOI, and the shape created when using solid green lines represents a slightly enlarged AOI with a buffer of 40 pixels.



(a) Rectangle AOI singular mapping (b) Rectangle AOI overlapped mapping

(c) Convex AOI singular mapping (d) Convex AOI overlapped mapping

FIGURE 9: Mapping eye fixation with different AOI types: the red "+" indicates the eye fixation location. For (b) and (d), we determine that an eye fixation occurred in all three AOIs.

where $MA_{I,T}$ is the set of mapped AOIs during a time frame and I is index.

Figure 9 represents a mapping example where the rectangular and convex AOIs are shown in green. The red "+" symbol represents the eye fixation point that falls within the AOI boundary. There may be situations when an eye fixation falls inside the boundary of more than one AOI simultaneously. In other words, the eye fixation falls into a region that is in the intersection of several AOI boundaries, thus giving rise to the concept of "overlapped AOI mappings." Thus, in this example, the mapped AOI set for this eye fixation will include three elements, which can be shown as $MA_{I,T} = \{a_{1,t}, a_{2,t}, a_{3,t}\}$.

Another important concept, which will be useful in the analysis, is the cardinality of the MA set, where cardinality is the number of elements present in that set. This can be expressed as follows:

$$\left|\mathrm{ma}_{i,t}\right| = c, \quad c \in \{0, 1, 2, 3, \ldots, n_t\}, \tag{7}$$

where $|\cdot|$ is the cardinality function and n_t is the number of multielement objects present at time t.

Thus, if "c" is the cardinality of the $\mathrm{ma}_{i,t}$ set we can say that the corresponding eye fixation index has been mapped to "c" number of AOIs simultaneously. The larger the cardinality of the $\mathrm{ma}_{i,t}$ set, the greater the difficulty in analyzing those eye

fixations. Therefore, an important consideration in the data analysis is the frequency distribution of different cardinal values of the $\mathrm{ma}_{i,t}$ set.

Step 4 (visualize plotted eye fixation data on the simulated scenarios). After the mapping process, the eye fixation data is overlaid on the simulated display as a function of time using the update rate. This process requires subsequently plotting both the eye fixations and AOI data pertaining to the same time frames and covering the time frames sequentially. Example cases are shown in Figure 12.

Step 5 (investigate the mapping effects for different AOI gap tolerance (AGT) values). Some of the metrics that are of particular interest for this study are (1) the "percentage of the number of eye fixations falling inside AOIs (PNFIA)" and (2) the "percentage of the duration of the eye fixations falling inside AOIs (PDFIA)."

PNFIA is defined as

$$\mathrm{PNFIA} = \frac{\text{Number of eye fixations falling inside AOIs}}{\text{Total number of eye fixations}}, \tag{8}$$

where the number of eye fixations falling inside the AOIs (in (8)) is

$$\text{Number of eye fixations falling inside AOIs} = \sum_{t=1}^{\max(T)} \sum_{i=1}^{m_t} A_{i,t}, \tag{9}$$

where $\max(T)$ is maximum value of the time frame count and m_t is the number of eye fixations during time frame t:

$$A_{i,t} = \begin{cases} 1 & \text{if } |\mathrm{ma}_{i,t}| \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where the cardinality function is expressed as $|\cdot|$ (e.g., $|\mathrm{ma}_{i,t}|$).

$A_{i,t}$ is the indicator function that becomes 1 if the cardinality of the corresponding set $\mathrm{ma}_{i,t}$ is nonzero; in other words this function takes the value of 1 if the associated eye fixation falls at least within one AOI boundary. Therefore, using (9) and (10) we get

$$\mathrm{PNFIA} = \frac{\sum_{t=1}^{\max(T)} \sum_{i=1}^{m_t} A_{i,t}}{N}, \tag{11}$$

where N is the total number of eye fixations.

PDFIA is defined as

$$\mathrm{PDFIA} = \frac{\text{Time duration of eye fixations falling within AOIs}}{\text{Total time duration of alleye fixations}}. \tag{12}$$

The time duration of eye fixations falling within AOIs is calculated as

$$D = \sum_{t=1}^{\max(T)} \sum_{i=1}^{m_t} d_{i,t}, \tag{13}$$

where $d_{i,t}$ is time duration of eye fixation index i during time frame t and m_t is the number of eye fixations that occurred during time frame t.

For the purpose of calculating the time duration of eye fixations falling within an AOI, we need to consider only those eye fixations indexes for which the cardinality of their corresponding AOI mapped set is nonzero. Therefore we can use the indicator function described in (10) to take into account only those specific eye fixation indexes that fall at least within one AOI boundary. Thus we get the following:

$$D' = \sum_{t=1}^{\max(T)} \sum_{i=1}^{m_t} A_{i,t} \times d_{i,t}. \tag{14}$$

Using (13) and (14) we get that the percent time duration of eye fixations falling within an AOI to be

$$\mathrm{PDFIA} = \frac{D'}{D}. \tag{15}$$

The next metric of interest is the frequency distribution of $\mathrm{ma}_{i,t}$ of various cardinalities. In other words, it is the frequency distribution of various possible "c" values, where c is as described in (7). This can be found by counting the number of occurrences of various possible values of "c." This frequency distribution is an important metric because it is a qualitative measure of the difficulty associated with the analysis of the eye fixation sequence.

Step 6 (change AOI gap tolerance (AGT) values). Due to the visual angle error, the choice of the AGT value depends on the discretion of the analyst. In absence of any established relationship between the AGT values and the relevant eye fixation parameters discussed in an earlier section, the optimal range of the AGT value becomes very much context dependent. As a result, it becomes important to study this relationship for the present context. Thus, the next step involves varying the AGT value to investigate its impact on the relevant metrics of interest. The equation governing the change in AGT can be written as

$$\mathrm{AGT}_{R+1} = \mathrm{AGT}_R + \delta, \tag{16}$$

where AGT_R is AOI gap tolerance value for the iteration value R and δ represents increments of AGT values (e.g., $\delta = 5$ pixels).

Table 2 shows the various values of the iteration counter r and the associated AGT values. All the above-mentioned steps need to be performed from Steps 2–5 for each r value.

Step 7 (find optimal AOI gap tolerance value). Assuming that a participant or a group of participants interrogate one object at a time, one method to find the optimal AGT value is to select the AGT value that provides the highest frequency of the mapped AOI set of cardinality 1, or in other words we can identify the optimal AGT value for which the number of eye fixations on single AOIs is maximum.

Table 2: AGT values defined for each iteration (r).

Various combination of r and AGT values	
r	AGT (pixels)
1	5
2	10
3	15
4	20
5	25
6	30
7	35
8	40
9	45
10	50
11	55
12	60
13	65
14	70
15	75
16	80
17	85
18	90
19	95
20	100

The equation to find the optimal AGT value ($\mathrm{AGT}_{\mathrm{optimal}}$) is as follows:

$$\mathrm{AGT}_{\mathrm{optimal}} = \underset{\mathrm{AGT}\in[5,10,\ldots,100]}{\arg\max} \left[\mathrm{freq}\,(c) : c = 1\right], \quad (17)$$

where c is cardinality of the mapped AOI set and $\mathrm{freq}(\cdot)$ is frequency of set with cardinality value c.

Note that we can also obtain an overall single near optimal AGT value recommended for an experiment if we used the aggregated eye tracking data obtained from multiple participants.

Pseudocode 1 shows the simplified pseudocode based on the algorithmic flowchart shown in Figure 5.

3. Implementation

The developed approach was benchmarked through retired professional air traffic control specialists (ATCSs) who primarily work as instructors for the Federal Aviation Administration (FAA). The experiment was held at the FAA Civil Aerospace Medical Institute (CAMI), located in Oklahoma City, OK.

3.1. Participants. Ten certified ATCSs with over 32 years of experience participated in the experiment. In addition, three FAA employees participated as pseudo pilots who maneuvered the aircraft based on the controllers' clearances. Eye tracking data were collected from the certified controllers. Due to the unforeseen technical issues when using the eye tracking system and the air traffic control simulator, the data obtained from the first five participants were discarded, and only the data obtained from the subsequent five participants were used.

3.2. Apparatus. The experiment environment closely resembled the actual environment in the field (Air Route Traffic Control Center) in order to obtain high face validity. The simulated air traffic scenarios were displayed using a 19.83 × 19.83-inch monitor (2048 × 2048-pixel active display area). The size and resolution were equivalent to the actual display size used in the field. An additional monitor was placed to the right of the simulation monitor to display the En Route Automation Modernization (ERAM) tool, a decision support tool that provides text data with respect to aircraft data, trajectory, and possible conflicts. A keyboard was placed beneath the simulation monitor for an ATCS to input commands.

The eye tracking data were only collected from the simulation monitor to test our designs and algorithms. Facelab 5 eye tracker system [11] was used to collect the eye tracking data with a sampling rate of 60 Hz. The threshold for defining a fixation was set at 100 ms. The accuracy of the eye tracker was in the range of 0.5°–1° of visual angle error. Each participant's eyes were approximately in the range of 55–70 cm from the simulated display. Kongsberg-Gallium I-Sim software, internally outsourced and used by the FAA, was used for generating three different air traffic scenarios. The refresh rate of the simulated radar display was 1 second. Obtained raw eye tracking data was exported through the Eyeworks software [21], and the data output was similar to that shown in Table 1.

The structure of the air traffic simulation file is provided in Table 3 (sample data). The output file contains the details of the aircraft movements, their coordinates, and other relevant details of the aircraft representation used for the simulation. The data update rate (UR) was 1 second. In Table 3, the first and second columns show the elapsed time from the start of the experiment and the actual time of day, respectively. The third column named "aircraft code" shows the code name of the aircraft under consideration. The fourth column is the "target" column which shows the horizontal (X pos) and vertical (Y pos) coordinates of the targets (aircraft) in pixels. The fifth column is the "data block" column which has three subparts: (1) top left corner coordinates of the data block, (2) bottom right coordinates of the data block, and (3) direction column that represents the relative location of the aircraft with respect to the target position (N (north), NE (northeast), E (east), SE (southeast), S (south), SW (southwest), W (west), and NW (northwest)). The last column provides the position coordinates in pixels of the vector line's end point.

3.3. Task and Scenarios. The task was a high fidelity representation of air traffic control as performed in the U.S. National Airspace System's Air Route Traffic Control Centers. Controlling simulated traffic such as this requires an experienced ATCS to observe the radar screen and give

```
for r = 1 till max(r) (loop to cover all iteration)
    for t = 1 till max(T) (loop to cover all time frames)
        for j = 1 till n_t (loop to cover all multi-element objects for the current time frame t)
            Plot jth plane elements for time frame t
            Plot jth AOI boundary for time frame t
        end for
        for i = 1 till m_t (loop to cover all eye fixation e_{i,t} for the current time frame t)
            Plot ith eye fixation (e_{i,t}) for time frame t
            for j = 1 till n_t (loop to check whether the current eye fixation falls within the AOI list of the current time frame t)
                find whether current fixation e_{i,t} falls inside AOI_{j,t}
                store the result: store 1 for inside, 0 for outside AOI
                store the time duration of the eye fixation
            end for
        end for
    end for
    calculate percent number of eye fixations within AOI
    calculate percent time duration of eye fixations within AOI
    calculate the frequency distribution of mapped AOI sets of various cardinalities
end for
calculate the optimal AGT value
```

Pseudocode 1: Pseudocode used for the overall process.

Table 3: Air traffic simulation sample output data.

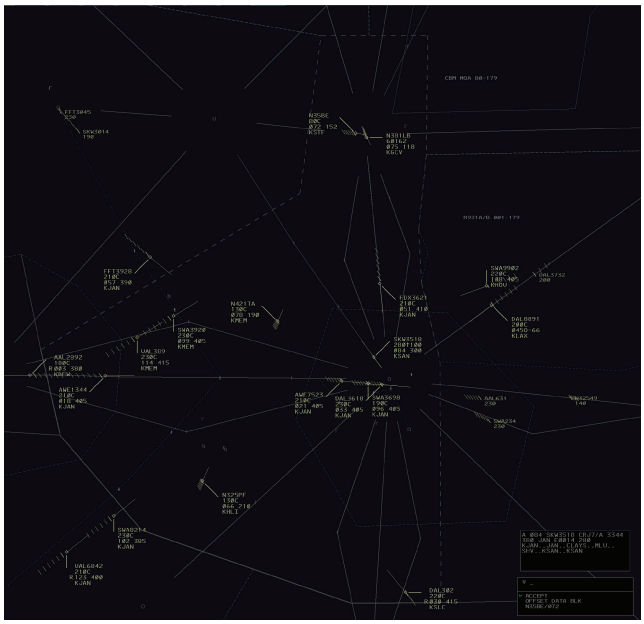
Scenario time	Time of the day	Aircraft code	Target		Data block				Direction	Vector line end point	
					Top left		Bottom right				
			X pos	*Y* pos	*X* pos	*Y* pos	*X* pos	*Y* pos		*X* pos	*Y* pos
00:00:14	11:55:19	DAL1268	988	939	1050	922	1141	994	E	957	892
00:00:14	11:55:19	EGF1819	599	1248	608	1282	699	1354	S	624	1237
00:00:14	11:55:19	N21LD	732	1300	792	1334	883	1406	SE	742	1277
00:00:15	11:55:20	DAL1268	987	938	1050	922	1141	994	E	957	892
00:00:15	11:55:20	EGF1819	599	1248	446	1231	537	1303	W	624	1237
00:00:15	11:55:20	N21LD	732	1299	792	1334	883	1406	SE	742	1277

Table 4: Characteristics of different simulation scenarios.

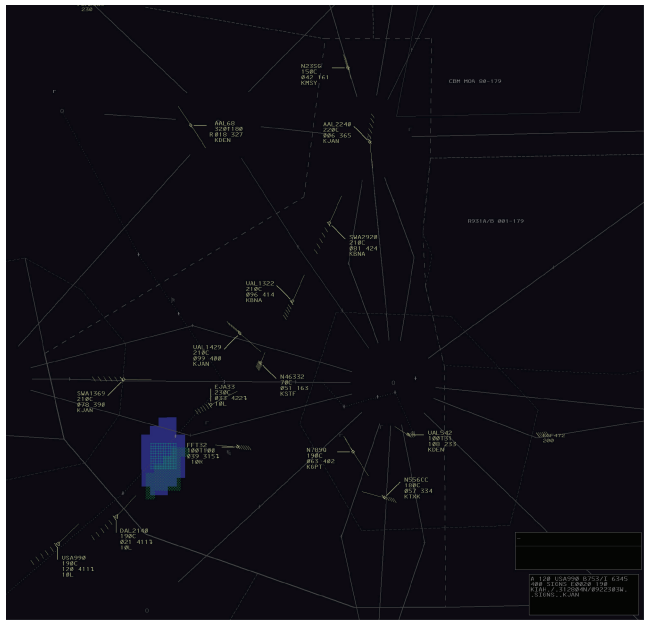
Scenario name	Average unique aircraft per frame	Min unique aircraft per frame	Max unique aircraft per frame	Std dev unique aircraft per frame
Moderate traffic	20	7	30	7
Moderate traffic + weather feature	20	6	29	6
Busy traffic	24	8	37	7

clearances to aircraft adjusting their altitudes, headings, or speeds so as to maintain aircraft-to-aircraft separation and route aircraft through the sector or to their destination airport within the sector. The ATCSs gave voice commands, via the communication system, to pseudo pilots who were situated in a remote room. The pseudo pilots followed the clearances and provided read-back to the ATCSs. Three scenarios were used (moderate traffic, moderate traffic with convective weather, and busy traffic). The duration of each scenario was 20 minutes. Table 4 and Figure 10 show the details of the scenarios. In Figure 10(b), the blue patch represents the weather feature.

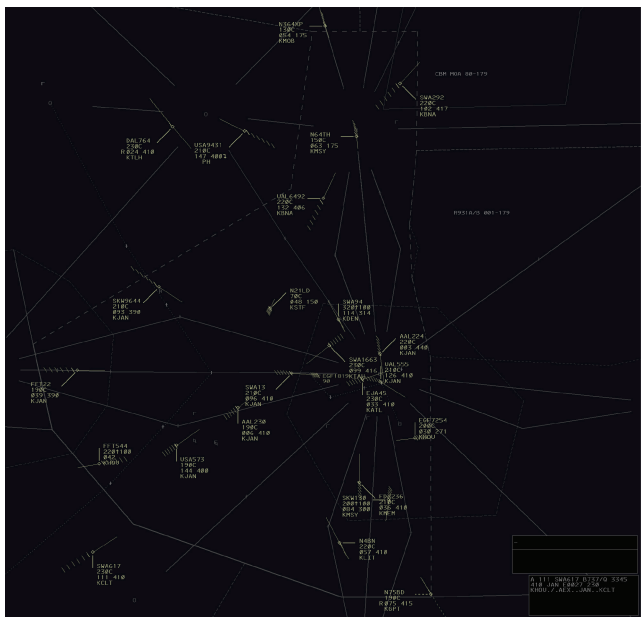
3.4. Data Analysis. The analysis of convex and rectangular AOIs was automated as follows: Based on the provided simulation output and the eye tracking output, both data sets were synchronized (step (1) in Figure 5). After the preprocessing steps, the two types of AOIs (convex and



(a) Moderate traffic scenario (Mod)



(b) Moderate traffic with weather feature scenario (Mod + W)



(c) Busy traffic scenario (Busy)

Figure 10: Air traffic control scenarios.

rectangular AOI) were created using the aircraft coordinates at every second (step (2)). Then, mapping was performed using the eye tracking data and the simulation data (step (3)). The mapped data was visualized (step (4)), and relevant metrics including the PNFIA and PDFIA were calculated by varying the AGT values (steps (5) and (6)). Finally, the optimal AGT value was obtained by identifying the highest percentage of the eye fixations on single AOIs (step (7)).

The complexity of the data processing time was $O(n_1 n_2 n_3 n_4 n_5 n_6)$, where n_1 is the number of participants, n_2 is the number of scenarios, n_3 is the number of AOI types, n_4 is the number of AGT values, n_5 is the number of AOIs per time frame, and n_6 is the number of eye fixations per time frame. Each eye fixation was compared with each AOI per time frame.

In the Results, the total eye fixation numbers and durations on the display (without using AOIs) were plotted in order to investigate the oculomotor trends. Then, aggregated PNFIA and PDFIA values for all participants were plotted based on the AGT values. Then, the number of eye fixations

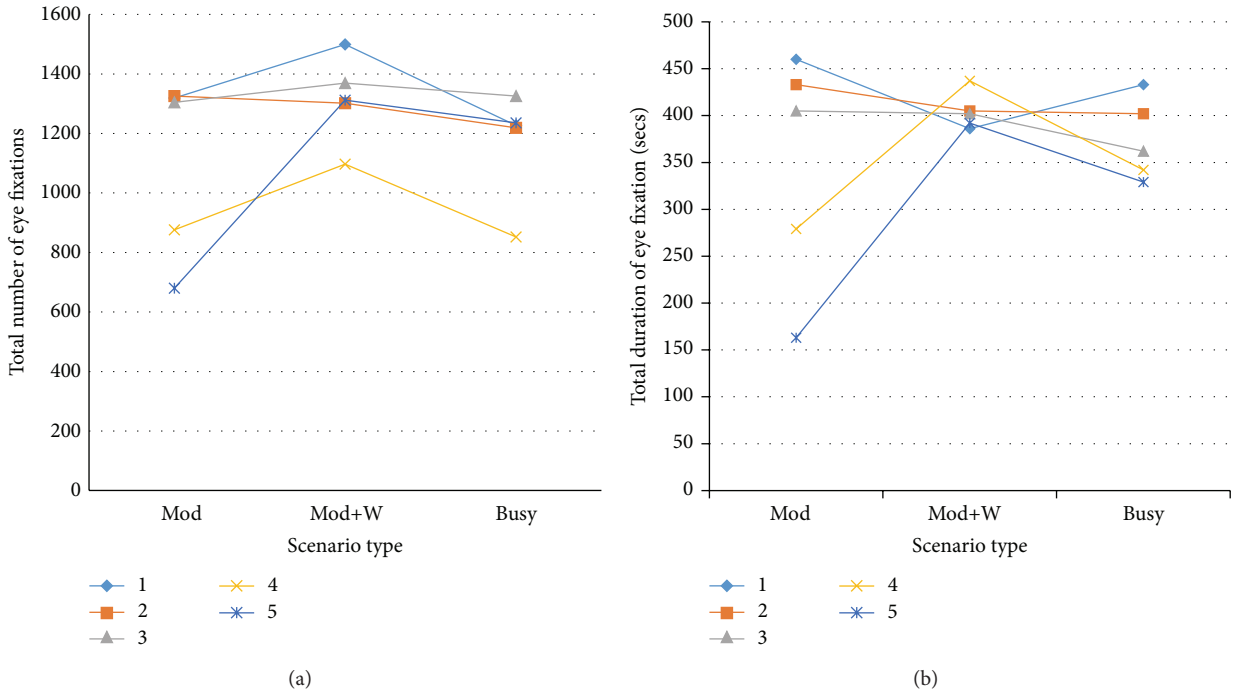


Figure 11: Oculomotor trends of the total number and duration of eye fixations among scenarios.

that occurred on single and multiple overlapping AOIs was plotted based on the AGT values. The optimal AGT value was computed, and examples of different scanpath sequences (resulting from either different AOI types or AGT values) were identified.

4. Results

The oculomotor trends are shown in Figure 11. Figure 11(a) shows the total number of eye fixations and Figure 11(b) shows the total duration of eye fixations with respect to scenario difficulties: moderate traffic (Mod), moderate traffic with weather feature (Mod + W), and busy traffic (Busy). The legends in Figure 11 showing 1, 2, 3, 4, and 5 represent the participant numbers.

Figure 12 displays example snapshots of the visualization process (see Step (4) in Figure 5) for both AOI types. The example snapshots show the dynamic AOIs with the AGT value set to 40 pixels. In Figure 12, the AOIs are highlighted in green and the order of eye fixations along with the associated saccades (connections between eye fixations when moving from one to the next) are highlighted in red. Note that the automated illustrations of the ordered eye fixations (shown in numbers) and the saccades linking the eye fixations are accumulated, meaning that the illustrations show all eye fixations from the scenario start time (time frame 1) until the indicated time frame such as time frame 120 or 1200.

Figure 13 depicts the effect of changing the AGT values on (1) the percentages of the numbers of eye fixations that fall within AOIs (PNFIA) shown in grey and (2) the percentages of the durations of the eye fixations that fall within AOIs (PDFIA) shown in black. The plots show the mean and standard error associated with every AGT value. In addition, the fitted polynomial equations and the R^2 values are provided.

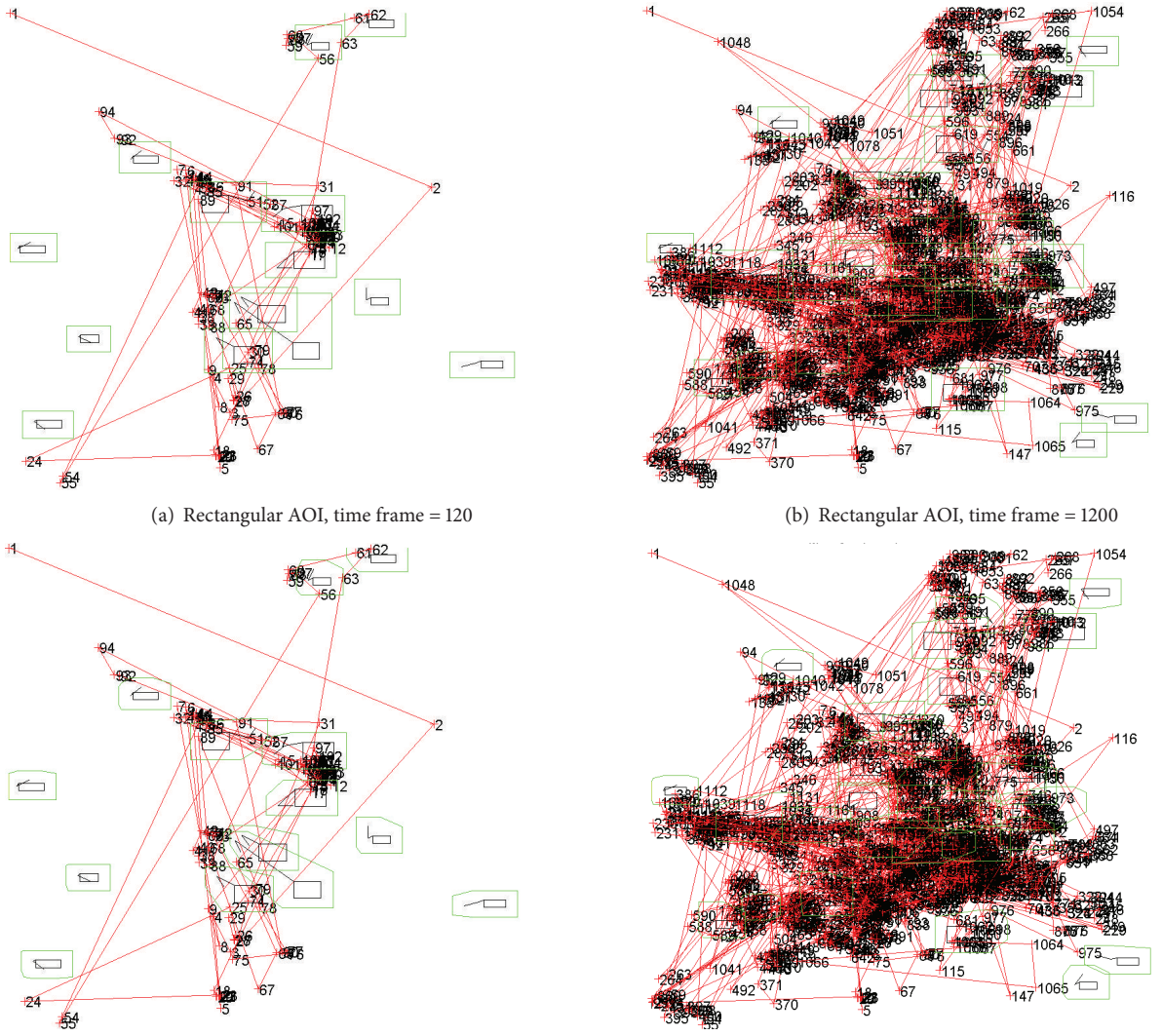
Table 5: Mean and standard error for the optimal AGT values for different AOI types.

Optimal AGT	AOI type	
	Convex AOI	Rectangular AOI
Mean (pixels)	40	38.3
Standard error (pixels)	1.8	1.2

Figure 14 depicts the change in the frequency of mapped AOI sets, of various cardinalities, with respect to the change in AGT values for convex and rectangular AOIs, respectively. The plots show the mean and the standard error associated with the coverage percent values. The maximum possible observed cardinality of the mapped AOI set is 8. A general trend among the various plots is that the frequency count of the $ma_{i,t}$ set having cardinality 1 (or in other words $c = 1$ (shown in red)) increased and then decreased. As the AGT values increased, the number of overlapping AOIs also increased, and the eye fixations on a single AOI subsequently decreased.

The near optimal (or recommended) AGT values (by considering all participants and scenarios) are provided in Table 5. The AGT value of 40 pixels captures approximately 70–80% of the total eye fixations that fall within the AOIs. Note that the participants can freely observe other areas that are not defined as AOIs within the display.

Figure 15 depicts the change in the frequency of mapped AOI sets, of various cardinalities, with respect to the change



(a) Rectangular AOI, time frame = 120

(b) Rectangular AOI, time frame = 1200

(c) Convex AOI, time frame = 120

(d) Convex AOI, time frame = 1200

Figure 12: Examples of visual representations of eye fixation data plotted onto the AOIs: the eye movements (eye fixation orders are numbered, and the saccadic movements are shown as red lines) were accumulated over time.

in AGT values for convex and rectangular AOIs, respectively. The plots show the mean and standard error associated with the frequency values for every AGT value. The maximum possible observed cardinality of the mapped AOI set is 8. In many cases the frequency of cardinality values higher than five was zero. Thus the curves for these cardinalities might not be exclusively visible on the plots as they are overlapping each other. A general trend among the various plots is that the frequency count of the $\mathrm{ma}_{i,t}$ set having cardinality 1 (or in other words $c = 1$ (shown in red)) increased and then decreased. As the AGT values increased, the number of overlapping AOIs also increased, and as a result, the eye fixations on a single AOI subsequently decreased.

Figure 16 shows examples of how different AGT values can affect the resulting AOI-based scanpath sequences. More relevant eye fixations were captured when using the optimal AGT value of 40 (obtained from our experiment) than the AGT value of 5. As shown in Figure 16, the identified scanpath sequence "FFCC(A,B)E" (Figure 16(b)) shows much more pertinent mappings compared to the scanpath sequence "CCA" (Figure 16(a)). Again, note that the scanpath sequences can be further collapsed into "FC(A,B)E" and "CA," respectively.

5. Discussion

An approach was developed that automatically (1) created rectangular and convex AOIs around multielement objects, (2) mapped eye fixations with different types of AOIs, (3) systematically evaluated the mapping characteristics by increasing the size of the AOIs to consider the fidelity of the eye trackers, and (4) investigated how the increase of the AOI sizes affects the overlapping of multiple AOIs. This approach was applied to the collection of visual scanning data from a

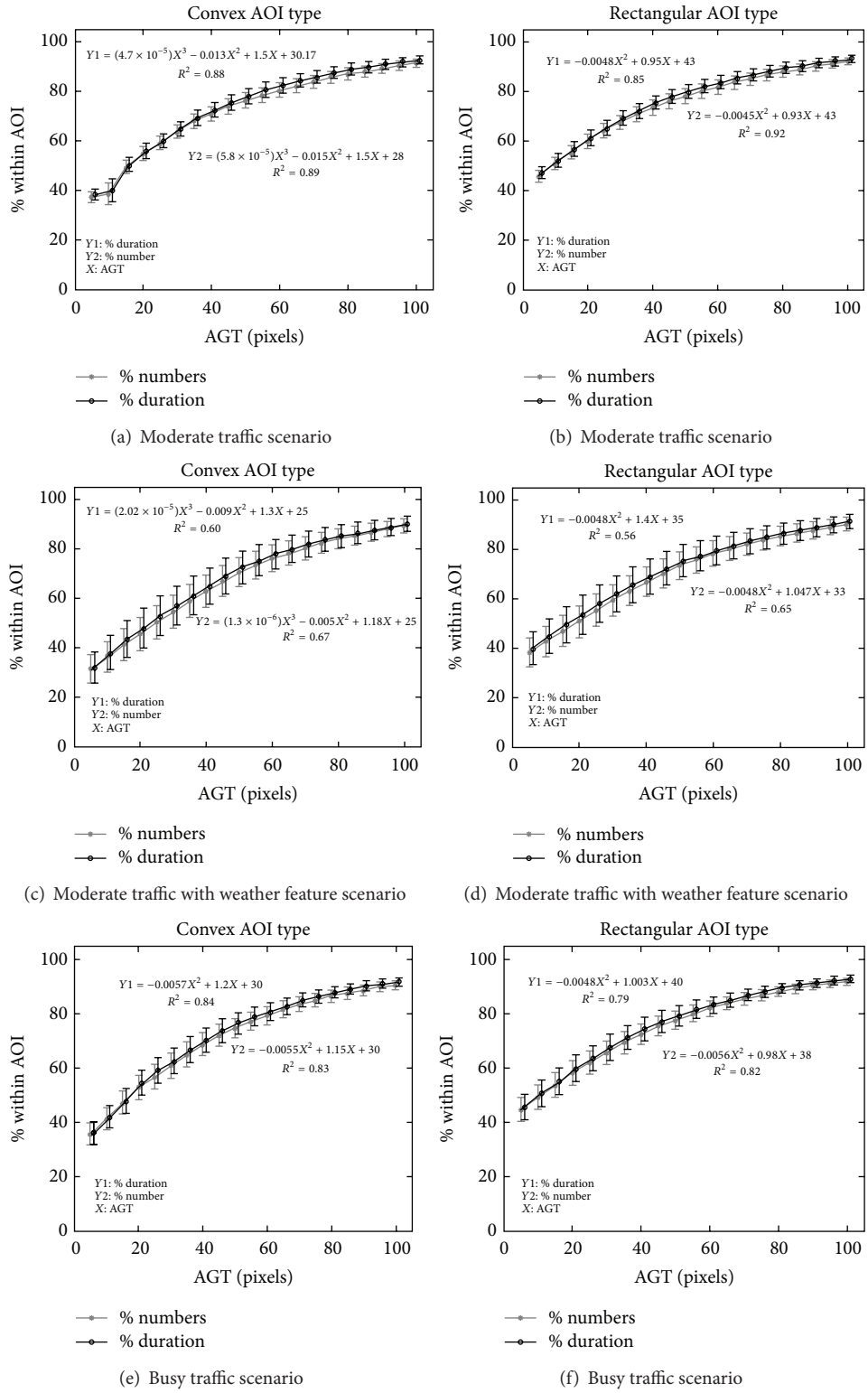
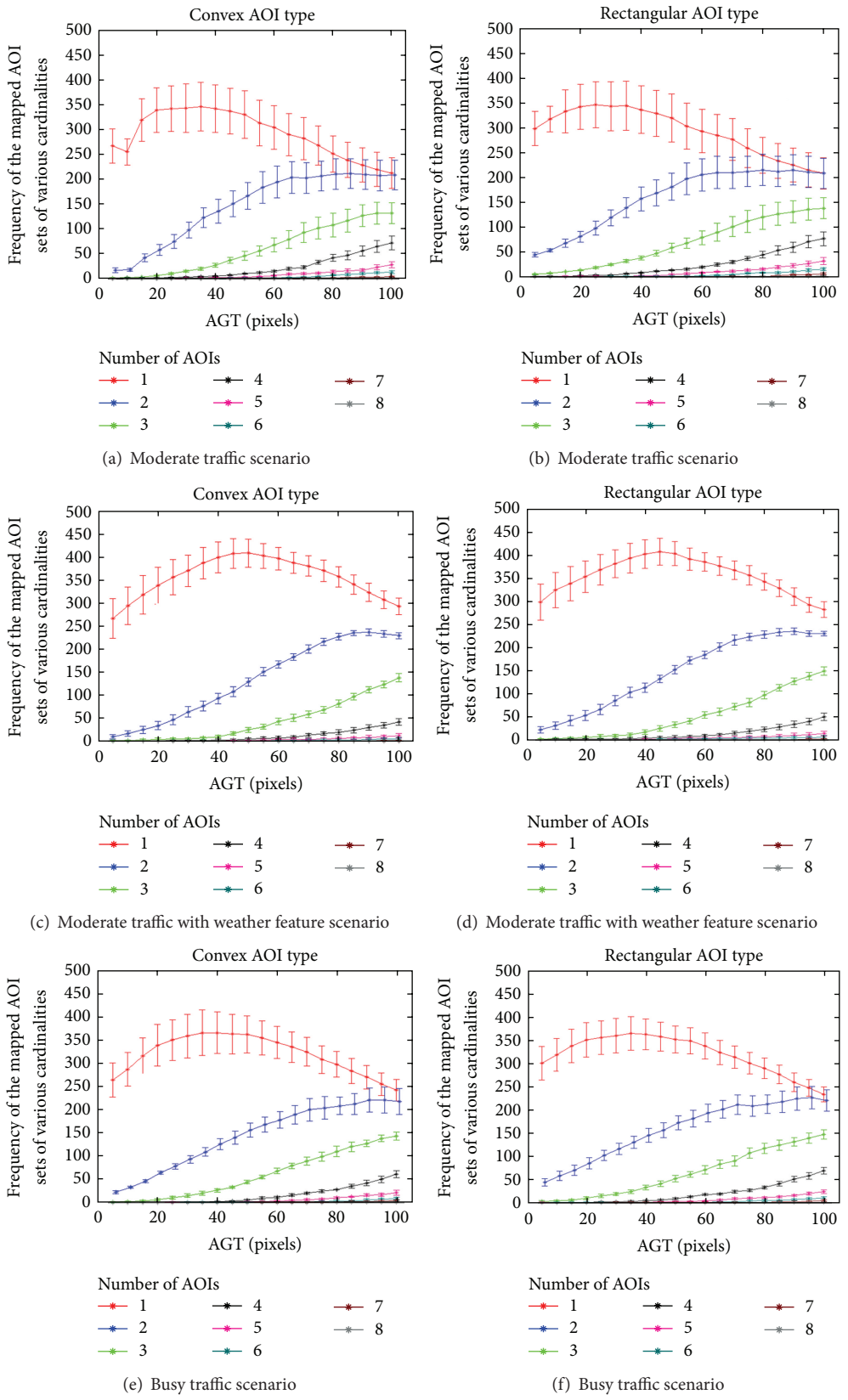


Figure 13: Plots of coverage percentages of the numbers and durations of the eye fixations that occurred within the AOI versus AGT values: the figures on the left column are the results for the convex type, and the figures on the right column are the results for the rectangular type.



(a) Moderate traffic scenario

(b) Moderate traffic scenario

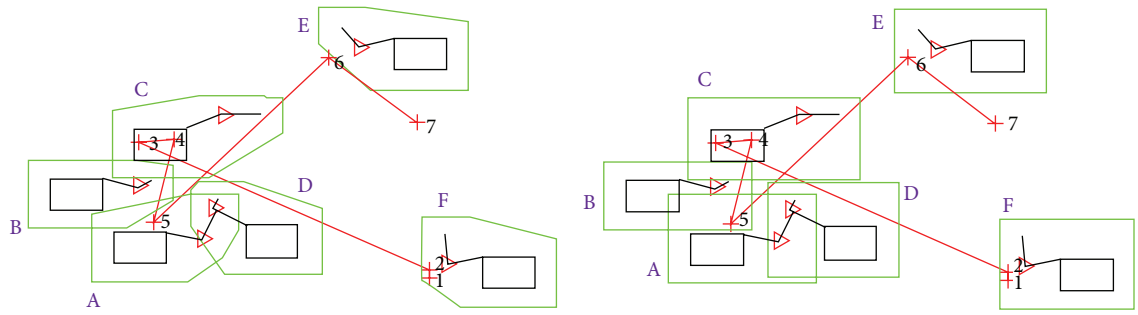
(c) Moderate traffic with weather feature scenario

(d) Moderate traffic with weather feature scenario

(e) Busy traffic scenario

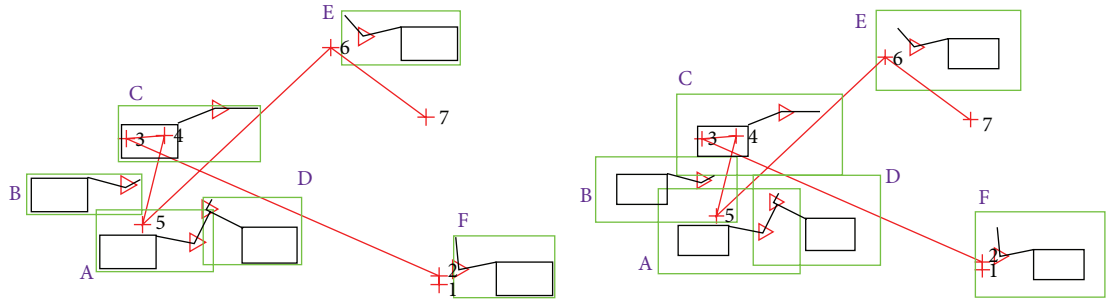
(f) Busy traffic scenario

Figure 14: Distribution of the number of eye fixations on single or overlapped AOIs based on AGT values: the top red line shows the change of the number of eye fixations for a single AOI. The subsequent lines show the change of the number of eye fixations on overlapping AOIs (increasing from 2 to 8).



(a) AGT = 40 (pixels) (b) AGT = 40 (pixels)

Figure 15: Examples illustrating how AOI types can affect the AOI-based scanpath sequences: the red "+" shows the location of the eye fixations, and the numbers are the corresponding eye fixation orders. For (a), eye fixation 5 only falls inside AOI "B," whereas for (b) eye fixation 5 falls inside both AOIs "A" and "B."



(a) AGT = 5 pixels (b) AGT = 40 pixels

Figure 16: Examples illustrating how AGT values can affect the AOI-based scanpath sequences: the red "+" shows the location of the eye fixations, and the numbers are the corresponding eye fixation orders. For (a), eye fixations 1, 2, 6, and 7 fall outside the AOIs, whereas for (b) only eye fixation 7 falls outside the AOIs.

high fidelity simulation of an air traffic control task. The task required ATCSs to interrogate multielement moving objects (that can change their overall shapes) on a radar display. The approach was applied to eye tracking data collected from the ATCSs as they performed the conflict detection and control task through interrogating multiple moving aircraft within their sector.

The oculomotor statistics on different types of scenarios show that the overall eye fixation numbers and durations on the display (without considering AOIs) did not significantly differ among the scenarios. The results differ from previous aircraft conflict detection research [22, 23]. In [22], eye fixation numbers and durations increased as the difficulty level increased (easy: many aircraft had different altitudes; moderate: many aircraft had similar altitudes; difficult: many aircraft changed altitudes), while setting the number of aircraft on the display at twelve for all scenarios. In [23], eye fixation numbers and durations increased as the number of aircraft on the display was increased from twelve to twenty. A major difference in the scenario settings was that there was no time limit on detecting possible collisions for [22, 23], whereas the experiment in this research had a time limit of twenty minutes.

Regarding the ATCSs' cognitive processes, one reason that similar oculomotor trends could be found is that the ATCSs were constantly vigilant on interrogating and controlling the aircraft throughout the experiment. In addition, the reason for a marginal decreasing trend on eye fixations and durations may be due to the order effect of the scenarios being performed in a sequence of moderate traffic, moderate traffic with convective weather, and busy traffic. The participants could have become more comfortable with the situation as they continued to control the multiple aircraft. Another possibility is that the ATCSs may have spent more time on looking at the ERAM display as well as the keyboard. Unfortunately, the exported eye tracking data only provides pixel-based eye fixations that occurred within the defined display; therefore, it is difficult to know where the eye fixations occurred outside the display.

The convex and rectangular AOI types did not generally affect the amount of mapped eye fixations among the participants and the scenarios due to the relatively small size of

the objects as well as the accuracy of the eye tracking system for a high face validity experiment. However, we were able to identify specific examples of different AOI types affecting the resulting scanpath sequence (Figure 15). The analysis of human performance using the scanpath sequences may have substantially differed for the same experiment if the analysts applied different AOI types. The effect may have been overall significant if the size of the multielement objects was bigger due to the increased unnecessary area (Figure 8) created by the rectangular AOI type. The unnecessary areas would also result in creating more overlapping AOI areas.

The AGT values substantially impacted the amount of covered eye fixations and durations on both AOI types and the trends fitted to polynomial equations. Up to a certain point, the increase of the AGT value was able to accommodate many eye fixations that occurred around the objects; then the increase rate (of the amount of included eye fixations) began to reduce since lower amount of eye fixations occurred further away from the objects. The eye fixation numbers and durations were highly correlated for our experiment. Note that the AGT values also affected the resulting scanpath sequences (Figure 16). The use of too tightly fitted AOIs resulted in missing many eye fixations that occurred around the object. Note that if we used AOIs that were too large, then the cardinality of the mapped AOI set would increase, leading to either inaccurate mapping or an increase in the complexity of the scanpath sequences by having more overlapping AOIs.

Thus, the selection of the AGT value gives rise to a trade-off between the coverage (amount of eye fixations) versus complexity (overlapping AOIs) of the algorithm because the more we increase the coverage, the more we increase the complexity. As the AGT value increases, the coverage of the overlapping AOIs increases accordingly, but the coverage of the single AOIs starts to decrease (Figure 14). The reason is that overlapping AOIs begin to take away the amount of eye fixations that occurred within single AOIs. Therefore, we were able to determine the near optimal AGT value by identifying the coverage peak of single AOIs. Having an adequate AOI size to map an eye fixation to a single AOI is more preferred to having larger AOIs that would result in creating unnecessary overlapping areas. In other words, the more we increase the coverage, the more we increase the complexity for multielement moving objects that can overlap.

6. Limitation and Future Research

Although the different AOI types did not show significant differences when aggregated results were compared, we were able to identify specific cases where differences were indeed present. A follow-up experiment is needed to vary the size of the actual objects in order to identify a threshold that shows substantial mapping differences when using complex convex approximations versus the simple shaped approximations. In addition, although the benchmarking of the developed methods was able to show that trade-offs exist when considering the design of AOIs based on visual angle errors and overlapping objects, more follow-up experiments are needed to refine and better support our methods.

In addition, the near optimal AGT values were obtained from aggregated data across the whole experiment and among the participants. The limitation to this approach is that we apply a constant AGT value for the whole duration. The optimal AGT value might not be a constant for all the time frames, and further detailed analysis might help to segregate time segments from the whole experimental duration (i.e., identify the amount of variations for different segregated time segments). Note that we would not be able to obtain a trend to identify the optimal value if the time length was too short (e.g., for a 1-second time frame, we would only obtain 1 or 2 eye fixations). To investigate how it would vary, we would first need to define the time segments that we should apply.

Another limitation is that we assumed that the multielement objects make discretized movements and that the scene (background) is fixed. If the background is moving or the objects make rapid movements (e.g., from one end of the screen to another end of the screen in a very short time), then our approach would not work. These issues are difficult to solve and should be addressed in our subsequent research.

The overarching goal of our research is to obtain more accurate mappings between the eye movements and the moving objects in order to better support the analysis of human performance. This research concentrated on prototyping, implementing, and evaluating new conceptual designs and algorithms to obtain more accurate mappings. Based on the obtained results in this research, we are currently analyzing the human performance based on the obtained AOI-based scanpath sequences through the Directed Weighted Networks [24, 25].

Furthermore, the results can be a basis to develop better scanpath analysis methods that build upon existing methods [22, 26–31], mimic human performance [32], and develop data visualization methods for active learning using the experts' visual scanning patterns [33]. In addition, the visual scanning data could be combined with EEG analysis [34] to better understand how the different types of tasks or incidents affect brain response and visual scanning and how the brain response data is correlated with visual scanning data.

7. Conclusion

To address the issue of mapping eye fixations with multielement objects (that move, can change their shape, and overlap over time), we proposed and implemented dynamic AOIs that represent the multielement objects. During the process, we showed a way to map eye fixations to overlapping AOIs. In addition, the concept of AGT was applied in order to address the issue of the fidelity of the eye trackers. Our approach was automated and applied to data collection from a high fidelity simulation of an air traffic control task. The benchmark showed that eye tracking data analyses can substantially differ based on how the AOIs are defined and how we can obtain near optimal values to better define the AOIs.

Competing Interests

There are no competing interests to declare.

Acknowledgments

This research was funded based on a cooperative agreement with the FAA NextGen Organization's Human Factors Division, ANG-C1 (Award no. 15-G-006) and conducted through collaboration with researchers at the Civil Aerospace Medical Institute's Aerospace Human Factors Division. The authors deeply appreciate the support from Dr. Carol Manning.

References

[1] M. A. Just and P. A. Carpenter, "Eye fixations and cognitive processes," *Cognitive Psychology*, vol. 8, no. 4, pp. 441–480, 1976.

[2] R. Pieters, E. Rosbergen, and M. Wedel, "Visual attention to repeated print advertising: a test of scanpath theory," *Journal of Marketing Research*, vol. 36, no. 4, pp. 424–438, 1999.

[3] C. Holland and O. V. Komogortsev, "Biometric identification via eye movement scanpaths in reading," in *Proceedings of the International Joint Conference on Biometrics (IJCB '11)*, pp. 1–8, Washington, DC, USA, October 2011.

[4] P. Konstantopoulos, P. Chapman, and D. Crundall, "Driver's visual attention as a function of driving experience and visibility. Using a driving simulator to explore drivers' eye movements in day, night and rain driving," *Accident Analysis and Prevention*, vol. 42, no. 3, pp. 827–834, 2010.

[5] G. Underwood, P. Chapman, N. Brocklehurst, J. Underwood, and D. Crundall, "Visual attention while driving: sequences of eye fixations made by experienced and novice drivers," *Ergonomics*, vol. 46, no. 6, pp. 629–646, 2003.

[6] P. Kasarskis, J. Stehwien, J. Hickox, A. Aretz, and C. Wickens, "Comparison of expert and novice scan behaviors during VFR flight," in *Proceedings of the 11th International Symposium on Aviation Psychology*, pp. 1–6, Columbus, Ohio, USA, March 2001.

[7] A. P. Tvaryanas, "Visual scan patterns during simulated control of an uninhabited aerial vehicle (UAV)," *Aviation Space and Environmental Medicine*, vol. 75, no. 6, pp. 531–538, 2004.

[8] K. Holmqvist, M. Nyström, and R. Andersson, *Eye Tracking*, OUP Oxford, Oxford, UK, 2011.

[9] Z. Kang and E. J. Bass, "Supporting the eye tracking analysis of multiple moving targets: design concept and algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '14)*, pp. 3184–3189, IEEE, San Diego, Calif, USA, October 2014.

[10] "Tobii Pro X2-60," Tobiipro.com, http://www.tobiipro.com/product-listing/tobii-pro-x2-60/.

[11] Ekstremmakina.com, "faceLAB 5—Seeing Machines," http://www.ekstremmakina.com/EKSTREM/product/facelab/index.html.

[12] "SensoMotoric Instruments GmbH, Gaze and Eye Tracking Systems, Products, RED250/RED 500," Smivision.com, http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/red250-red-500.html.

[13] M. Burke, A. Hornof, E. Nilsen, and N. Gorman, "High-cost banner blindness: Ads increase perceived workload, hinder visual search, and are forgotten," *ACM Transactions on Computer-Human Interaction*, vol. 12, no. 4, pp. 423–445, 2005.

[14] S. Mandal and Z. Kang, "Eye tracking analysis using different types of Areas of Interest for multi-element moving objects: results and implications of a pilot study in air traffic control," in *Proceedings of the Human Factors and Ergonomics Society 59th Annual Meeting*, pp. 1515–1519, Los Angeles, Calif, USA, 2015.

[15] D. Crawford, D. Burdette, and W. Capron, "Techniques used for the analysis of oculometer eye-scanning data obtained from an air traffic control display," Tech. Rep., NASA, 1993.

[16] E. Stein, *Air Traffic Controller Scanning and Eye Movements in Search of Information—A Literature Review*, Federal Aviation Administration Technical Center Atlantic, 1989.

[17] B. Willems, R. Allen, and E. Stein, *Air Traffic Control Specialist Visual Scanning II: Task Load, Visual Noise, and Intrusions into Controlled Airspace*, Federal Aviation Administration Technical Center, Atlantic City, NJ, USA, 1999.

[18] P.-V. Paubel, P. Averty, and E. Raufaste, "Effects of an automated conflict solver on the visual activity of air traffic controllers," *International Journal of Aviation Psychology*, vol. 23, no. 2, pp. 181–196, 2013.

[19] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.

[20] "MathWorks. Mapping toolbox," http://www.mathworks.com/products/mapping/.

[21] Eyetracking.com, "Powerful eye tracking software developed for researchers," http://www.eyetracking.com/Software/EyeWorks/.

[22] Z. Kang and S. J. Landry, "An eye movement analysis algorithm for a multielement target tracking task: maximum transition-based agglomerative hierarchical clustering," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 13–24, 2015.

[23] S. N. McClung and Z. Kang, "Characterization of visual scanning patterns in air traffic control," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 8343842, 17 pages, 2016.

[24] M. Tory, M. S. Atkins, A. E. Kirkpatrick, M. Nicolaou, and G.-Z. Yang, "Eyegaze analysis of displays with combined 2D and 3D views," in *Proceedings of the IEEE Visualization Conference (VIS '05)*, pp. 519–526, Minneapolis, Minn, USA, October 2005.

[25] M. Saptarshi, Z. Kang, J. Crutchfield, and A. Millan, "Data visualization of complex eye movements using directed weighted networks: a case study on a multi-element target tracking task," in *Proceedings of the 60th Annual Meeting of the Human Factors and Ergonomics Society*, Washington, DC, USA.

[26] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu, "Sequential pattern mining using a bitmap representation," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 429–435, ACM, July 2002.

[27] A. Çöltekin, S. I. Fabrikant, and M. Lacayo, "Exploring the efficiency of users' visual analytics strategies based on sequence analysis of eye movement recordings," *International Journal of Geographical Information Science*, vol. 24, no. 10, pp. 1559–1575, 2010.

[28] F. Cristino, S. Mathôt, J. Theeuwes, and I. D. Gilchrist, "ScanMatch: a novel method for comparing fixation sequences," *Behavior Research Methods*, vol. 42, no. 3, pp. 692–700, 2010.

[29] R. Dewhurst, M. Nyström, H. Jarodzka, T. Foulsham, R. Johansson, and K. Holmqvist, "It depends on how you look at it: scanpath comparison in multiple dimensions with MultiMatch, a vector-based approach," *Behavior Research Methods*, vol. 44, no. 4, pp. 1079–1100, 2012.

[30] J. Goldberg and J. Helfman, "Scanpath clustering and aggregation," in *Proceedings of the ACM Symposium on Eye-Tracking Research and Applications (ETRA '10)*, pp. 227–234, Austin, Tex, USA, March 2010.

[31] S. Mathôt, F. Cristino, I. D. Gilchrist, and J. Theeuwes, "A simple way to estimate similarity between pairs of eye movement sequences," *Journal of Eye Movement Research*, vol. 5, no. 1, article 4, 2012.

[32] Z. Kang and S. J. Landry, "Top-down approach for a linguistic fuzzy logic model," *Cybernetics and Systems*, vol. 45, no. 1, pp. 39–55, 2014.

[33] Z. Kang and S. J. Landry, "Using scanpaths as a learning method for a conflict detection task of multiple target tracking," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 56, no. 6, pp. 1150–1162, 2014.

[34] A. N. Belkacem, S. Saetia, K. Zintus-Art et al., "Real-time control of a video game using eye movements and two temporal EEG sensors," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 653639, 10 pages, 2015.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 2381451, 12 pages
http://dx.doi.org/10.1155/2016/2381451

Research Article

Learning to Model Task-Oriented Attention

Xiaochun Zou,[1] Xinbo Zhao,[2] Jian Wang,[2] and Yongjia Yang[2]

[1]*School of Electronics and Information, Northwestern Polytechnical University, Chang'an Campus, P.O. Box 886, Xi'an, Shaanxi 710129, China*
[2]*School of Computer Science, Northwestern Polytechnical University, Chang'an Campus, P.O. Box 886, Xi'an, Shaanxi 710129, China*

Correspondence should be addressed to Xinbo Zhao; xbozhao@nwpu.edu.cn

Received 27 November 2015; Accepted 28 March 2016

Academic Editor: Francesco Camastra

For many applications in graphics, design, and human computer interaction, it is essential to understand where humans look in a scene with a particular task. Models of saliency can be used to predict fixation locations, but a large body of previous saliency models focused on free-viewing task. They are based on bottom-up computation that does not consider task-oriented image semantics and often does not match actual eye movements. To address this problem, we collected eye tracking data of 11 subjects when they performed some particular search task in 1307 images and annotation data of 2,511 segmented objects with fine contours and 8 semantic attributes. Using this database as training and testing examples, we learn a model of saliency based on bottom-up image features and target position feature. Experimental results demonstrate the importance of the target information in the prediction of task-oriented visual attention.

1. Introduction

For many applications in graphics, design, and human computer interaction, it is essential to understand where humans look in a scene with a particular task. For example, an understanding of task-oriented visual attention is useful for automatic object recognition [1], image understanding, or image search [2, 3]. It can be used to direct visual search and foveated image video compression [4, 5] and robot localization [6, 7]. It can also be used in advertising design or implementation of smart cameras [8].

However, it is not easy to simulate task-oriented human visual behavior perfectly by machine. Attention is an abstractive concept, and it needs objective metrics for evaluation. Judging the results of experiments by intuitive observation is not precise because different people might focus on different regions of the same scene, even with task. To solve this issue, eye tracker equipment pieces that can record human eye fixation, saccades, and gazes are routinely used. Investigations of human eye movement data provide more objective ground truth for studies on computational attention models. At the present time, there are over two dozen databases with eye tracking data for both images and videos in the public domain [9], which mainly focus on "free-viewing" eye movements.

Most existing computational visual attention saliency models have often been evaluated against predicting human fixations in free-viewing task, in which some are biologically inspired and based on a bottom-up computational model and others combine both bottom-up image based saliency cues and top-down image semantic dependent cues. Though the models do well qualitatively, the models have limited use because they frequently perform well only in the context-free scenario.

Motivated by this, we make two contributions in this paper. The first is a large database of task-oriented eye tracking experiments with labels and analysis and the second is a supervised learning model of saliency which combines both bottom-up image based saliency cues and task-oriented image semantic dependent cues. Our database consists of eye tracking data from 11 different users across 1307 images. To our knowledge, it is the first time that such an extensive collection of task-oriented eye tracking data is available for quantitative analysis. For a given image, the eye tracking data is used to create a "ground truth" saliency map which represents where viewers actually look with a particular search task. We introduce a set of bottom-up image features and target position features to define salient locations and use a linear support vector machine to train a model of saliency.

We compare the performance of saliency models created with different task-oriented attention and show that our approach performs better in predicting human visual attention regions than MIT model [3], which is one of the best models in predicting context-free human gaze.

The structure of this paper is as follows: Section 2 provides a brief description and discussion of some previous works. Section 3 is devoted to describing the characteristics of the database. In Section 3.1, we present the data collection method, the images, eye tracking data, and ground truth data. Section 3.2 analyzes the properties of our database. The detailed description of our model is in Section 4 that evaluates our approach using the popular saliency model evaluation scores (AUC) with MIT saliency model. The discussion and conclusions are discussed in the last section.

2. Related Work

Attention and saliency play important roles in visual perception. In past few years, more than two dozen of such databases are now available in the public domain. Fixations in Faces (FIFA) [10] were collected from eight subjects performing a 2 s long free-viewing task on 180 color natural images. It demonstrates the fact that faces attract significant visual attention. Subjects were found to fixate on faces with over 80% probability within the first two fixations. The NUSSEF database [11] was compiled from a pool of 758 images and 75 subjects. Each image was presented for 5 seconds and free-viewed by at least 13 subjects. A big feature of this dataset compared with others is that the 758 images in the dataset contain a large number of semantically affective objects/scenes such as expressive faces, nudes, unpleasure concepts, and interactive actions. MIT database from Judd et al. [12] included 1003 images collected from Flickr and LabelMe. Eye movement data were recorded from 15 users who free-view these images for 3 s. In this database, fixations were found around faces, cars, and text. Many fixations are biased towards the center. The DOVES dataset [13] includes 101 natural grayscale images [14]. Eye movements from 29 human observers as they free-view the images were collected. However, all of these databases record "free-viewing" eye movements. In addition, MIT CVCL Search Model Database [15] was recorded to understand task-oriented eye movement patterns of users. Observers were asked to perform a person detection task, and their eye movements were found to be consistent, even when the target was absent from the scene. This database was recorded based on task-oriented attention, but its task is single. So it is necessary to create a content-rich database based on task-oriented attention.

Several visual attention models are directly or indirectly inspired by cognitive concepts which are from psychological or neurophysiological findings. The winner-take-all (WTA) biologically plausible architecture which is related to the Feature Integration Theory is proposed by Koch and Ullman [16]. Built on WTA, Itti et al. [17] first implemented the computational model using a center-surround mechanism and hierarchical structure to predict salient regions. In this model, an image is predecomposed into low-level attributes such as color, intensity, and orientation across several spatial scales. The WTA inference pulls out the position with most conspicuity set of features. Later, Le Meur et al. [18] proposed a bottom-up coherent computational approach based on the structure of the human visual system (HVS), which used contrast sensitivity, perceptual decomposition, visual masking, and center-surround interaction techniques. It extracted features in Krauskopf's color space and implemented saliency in three separate parallel channels: visibility, perceptual grouping, and perception. A feature map is obtained for each channel, and then a unique saliency map is built from the combination of those channels. Based on the isotropic symmetry and radial symmetry operators of Reisfeld et al. [19] and the color symmetry of Heidemann [20], Kootstra et al. [21] developed three symmetry-saliency operators and compared them with human eye tracking data. E. Erdem and A. Erdem [22], Marat et al. [23], and Murray et al. [24] are other models guided by cognitive findings.

Another class of models is derived mathematically. Itti and Baldi [25] defined surprising stimuli as those which significantly change beliefs of an observer. This is modeled in a Bayesian framework by computing the KL divergence between posterior and prior beliefs. Similarly, Zhang et al. [26] proposed SUN (Saliency Using Natural statistics) model in which bottom-up saliency emerges naturally as the self-information of visual features. Bruce and Tsotsos [27] present a model for visual saliency built on a first principles information theoretic formulation dubbed Attention based on Information Maximization (AIM). Avraham and Lindenbaum's work on Esaliency [28] uses a stochastic model to estimate the most probable targets mathematically. Schölkopf et al. [29] proposed the Graph-Based Visual Saliency (GBVS) model, which used a Markovian approach to describe dissimilarity and concentration mass regions. Seo and Milanfar [30] and Liu et al. [31] are two other methods based on mathematical models.

Another class of models computes saliency in the frequency domain. Hou and Zhang [32] proposed Spectral Residual Model (SRM) by relating spectral residual features in spectral domain to the spatial domain. In [28], Avraham and Lindenbaum proposed Esaliency, a stochastic model, to estimate the probability of interest in an image. They roughly segmented the image first and used a graphical model approximation in global considerations to determine which parts are more salient.

Our proposed approaches are related to those models that learn mappings from recorded eye fixations or labeled salient regions. These models use some high-level features obtained from earlier databases and conduct learning mechanisms to determine model parameters. Torralba et al. [33] proposed an attentional guidance approach that combines bottom-up saliency, scene context, and top-down mechanisms to predict image regions likely to be fixated by humans in real-world scenes. Based on a Bayesian framework, the model computes global features by learning the context and structure of images, and the top-down tasks can be implemented in the scene priors. Cerf et al. [34] proposed a model that adds several high-level semantic features such as faces, text, and objects to predict human eye fixations. Judd et al. [12] proposed a learning-based method to predict saliency.

They used 33 features including low-level features such as intensity, color, and orientation; midlevel features such as a horizon line detector; and high-level features such as a face detector and a person detector. The model used a support vector machine (SVM) to train a binary classifier. Zhao and Koch [35] proposed a model similar to that of Itti et al. [17], but with faces as an extra feature. Their model combines feature maps with learned weighting and solves the minimization problem using an active set method. Among the models described above, some focus on adding high-level features to improve predictive performance, while others use machine learning techniques to clarify the relationship between features and their saliency. However, the so-called high-level features are blur concepts and do not encompass all types of environments.

These saliency models have been used to characterize RoIs in free-viewing task, but their use in particular task has remained very limited. Recent results suggest that, during task-oriented visual attention, in which subjects are asked to find a particular target in a display, top-down processes play a dominant role in the guidance of eye movements [36–40]. However, the so-called top-down features are blur concepts and do not encompass all types of environments. Here, we exploit more informative concepts including low-level, target location, and center bias, using machine learning for eye fixation prediction.

3. Database of Eye Tracking Data

We collected a large database of eye tracking data to allow large-scale quantitative analysis of fixation points and gaze paths and to provide ground truth data for saliency model research [41]. Compared with several eye tracking datasets that are publicly available, the main motivation of our new dataset is for studying task-oriented visual attention, that is, where observers look while deciding whether a scene contains a target.

3.1. Data Gathering Protocol

3.1.1. Participants. Fifteen participants, undergraduate and graduate volunteers aged 19–32 years ($\mu = 23.3$, $\sigma = 38.4$) with uncorrected and corrected normal eyesight, voluntarily joined this experiment. All the participants were from the Northwestern Polytechnical University.

3.1.2. Apparatus. Tobii TX300 eye tracker was used to record eye movements. We set the sampling frequency to 300 Hz. The eye tracker tolerates a certain extent of head movements, which allows the subjects to move freely and naturally in front of the stimulus. Freedom of head movement is at 65 cm, 37 × 17 (width × height), where at least one eye is within the eye tracker's field of view. Max head movement speed 50 cm/s stimuli were presented on a 23-inch wide screen TFT monitor. The screen size was 50.5 cm × 28.5 cm. Its screen response time was typically 5 ms and its resolution was set to 1920 × 1080.

3.1.3. Materials. We randomly selected 1307 images from VOC2012 as the stimuli. The longest dimension (could be either width or height) of each image was 500 pixels and the other dimension ranged from 213 to 500 pixels. The images contained eight categories, namely, airplane, motorbike, bottle, car, chair, dog, horse, and person.

3.1.4. Procedure. The 1307 images were separated into eight groups. Each group contained 100 images from the same categories and 70 images from the other categories (10 images were selected from each of these categories). All subjects sat at a distance of approximately 65 cm from the screen in a relatively quiet room. The images from each group were presented randomly with their original size in the middle of screen. Before the test, a five-point target display was used for calibration. To ensure high-quality tracking results, we checked the calibration accuracy after each of the groups. If the accuracy of the eye tracker was within about 1° visual angle, the subjects can continue the next group. Otherwise, the calibration will be carried out again. Subjects will be given different instructions for each of the groups. For example, for airplane group, subjects would be asked to find airplane in each picture, while a picture may have zero, one, or more airplanes. Subjects should find airplanes as more as possible in one image and switch to the next one through hitting the space key. To encourage the subjects to concentrate on looking for the target, we took two measures to improve authenticity of test. On the one hand, each group (above-mentioned eight groups) was equally divided into three small subsets. Subjects will spend less time to view the small subsets and pay more attention to the stimuli. On the other hand, after each subset, the subjects took a 2 min break and did a memory test: how many airplanes did you find?

3.2. Analysis of Dataset

3.2.1. Consistency. In our dataset, for the target-present images, all subjects fixate on the same locations, while, in target-absent image, subjects' fixations are dispersed all over the image. We analyze this consistency of human fixations over an image by measuring the entropy of the average continuous saliency map across subjects. Though the original images were of varying aspect ratios, we resized them to 200 × 200 pixel images before calculating entropy. Figure 1(c) shows a histogram of the entropies of the images in our database. It also shows a sample of 12 saliency maps (shown in Figures 1(a) and 1(b)) with lowest and highest entropy and their corresponding images.

3.2.2. Center Bias. Our data indicates a strong bias for human fixations to be near the center of the image, as is consistent with previously analyzed eye tracking datasets [12, 42]. Figure 2 shows the average human saliency map separately from the dog and chair category, which have the strongest and weakest center bias. In the dog category, 57% of the gaze points lie within the center 11% of the image, and 80% of the gaze points lie within the center 25% of the image. In the chair category, 29% of the gaze points lie within the

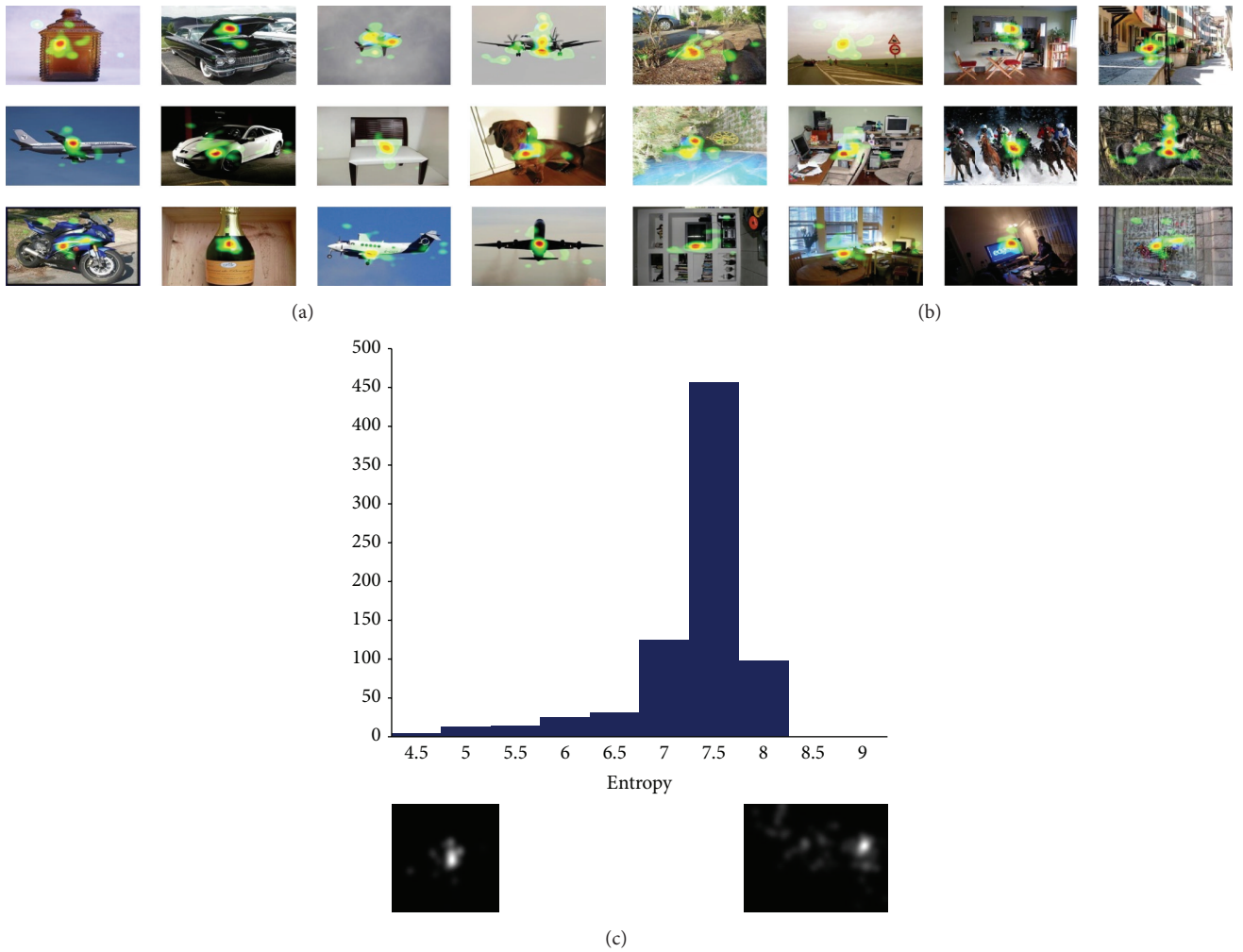


Figure 1: ((a) and (b)) The heat map made from subjects gaze points with low and high entropy. If the image has high entropy, it usually contains more objects. (c) A histogram of the saliency map entropies.

center 11% of the image, and 49% of the gaze points lie within the center 25% of the image.

There are several hypotheses for the root cause of center bias. In our test, the main reason is that people tend to place object or interesting things near the center of an image when taking a picture (the so-called photographer bias). To test this notion, we separately analyze percent of target gaze points, which are gaze points located on the target object within the center 11% and 25% of the dog and chair category. Obviously, in the dog category percentage of target gaze points in center area is more than that in the chair category. This difference has been attributed to the fact that target object mainly located on the center of images in dog category but was distributed in the whole image in chair category.

3.2.3. Agreement among Observers. In this paragraph, we evaluate agreement of the fixation positions among observers. Analysis of the eye movement patterns across observers showed that the fixations were strongly constrained by the search task and the scene context. To evaluate quantitatively the agreement among observers, we studied the human interobserver (IO) model to predict eye fixations, under the same experimental conditions. The IO model outputs, for a given stimulus, a map built by integrating eye fixations from subjects other than the one under test while they watched that stimulus. Then the map was used to predict fixations of the excluded subject. Finally, we use the evaluation of IO model performance to evaluate the agreement among observers.

Using the area under the ROC curve (AUC) as the score, the IO model's map is treated as a binary classifier on every pixel in the image. Pixels with larger values than a threshold are classified as fixated while the rest of pixels are classified as nonfixated. Human fixations are used as ground truth. By varying the threshold, the ROC curve is drawn as the false positive rate versus true positive rate, and the area under this curve indicates how well the saliency map predicts actual human eye fixations.

We separately computed the IO model over 8 categories from our dataset and select the mean value as the result. Table 1 shows the mean value of AUC scores of models.

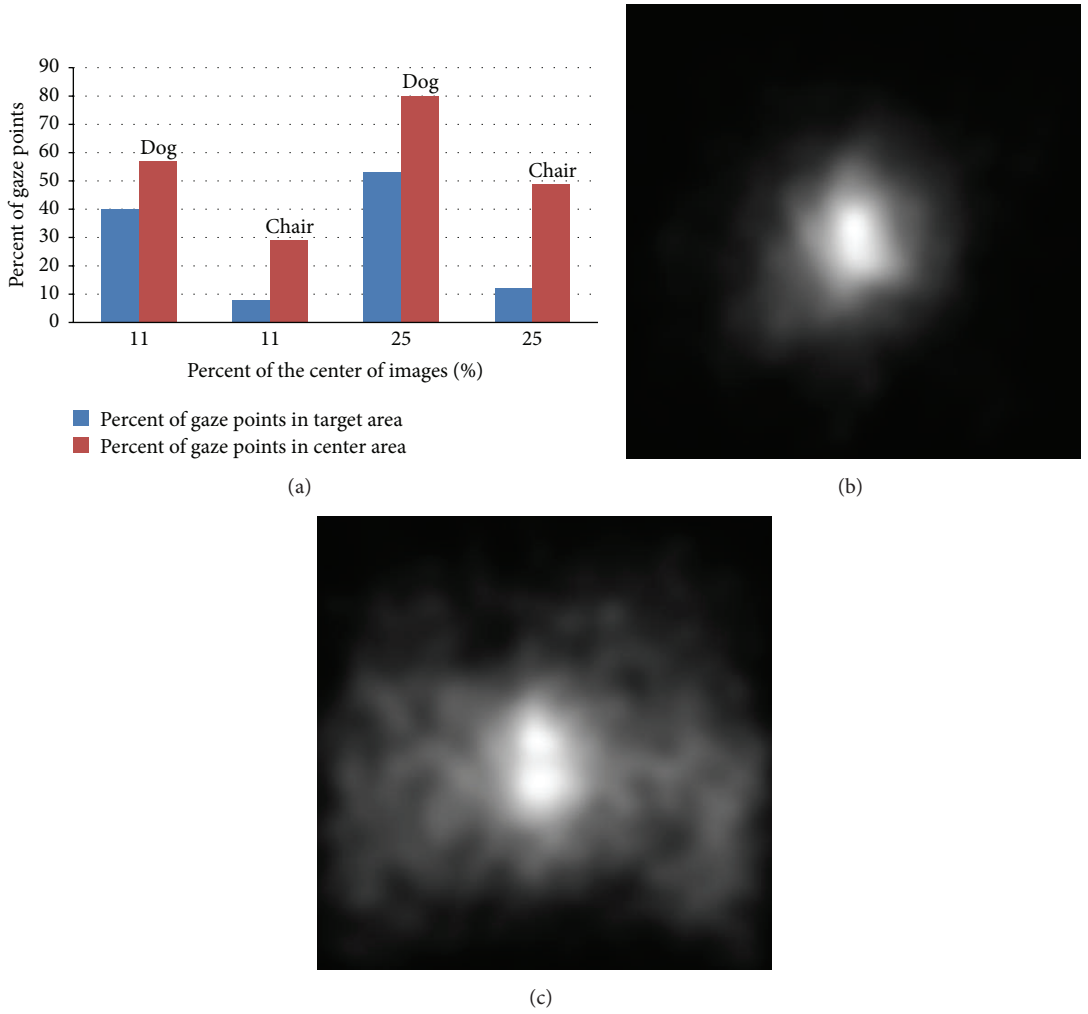


Figure 2: (a) The percentage of gaze points within the center 11% and 25% of the images, which is displayed by blue. Meanwhile, red shows the percentage of target gaze points. Obviously, in the dog category, the percentage of target gaze points is more than chair category. ((b) and (c)) Dog's and chair's average saliency map containing all the gaze points, which indicates a bias to the center of the image.

Table 1: Intersubject agreement for target-present and target-absent.

Group name	Target-present	Target-absent
Airplane	0.90	0.90
Bottle	0.87	0.87
Car	0.86	0.86
Chair	0.83	0.84
Dog	0.95	0.95
Horse	0.94	0.94
Motorbike	0.93	0.93
Person	0.92	0.93
Average	0.90	0.92

The results show that observers are very consistent with one another on the fixated locations in the target-present and target-absent conditions (over 85% in each case). On average, the agreement among observers is higher when the target is present than absent. This suggests that locations fixed by observers in target-present image are driven by the target location.

3.2.4. Gaze Points in Each Stimulus. The task of counting target objects within picture is similar to an exhaustive visual search task. In our design, each scene could contain up to 4 targets. Target size was not prespecified and varied among the stimuli set. Under these circumstances, we expected observers to exhaustively search each scene, regardless of the true number of targets present. Figure 3 shows the average number of the total of gaze points of each stimulus in every group. Unexpectedly, the count of fixations in the target-present is obviously more than target-absent.

To analyze the fixation position in the target-present images, we compare the percentage of human fixation that falls within the target object and the center area. In the first case, we apply the ground truth segmentation as the target object's area. In the second case, we calculate the percentage of human fixations located within the center 2%, 11%, 25%,

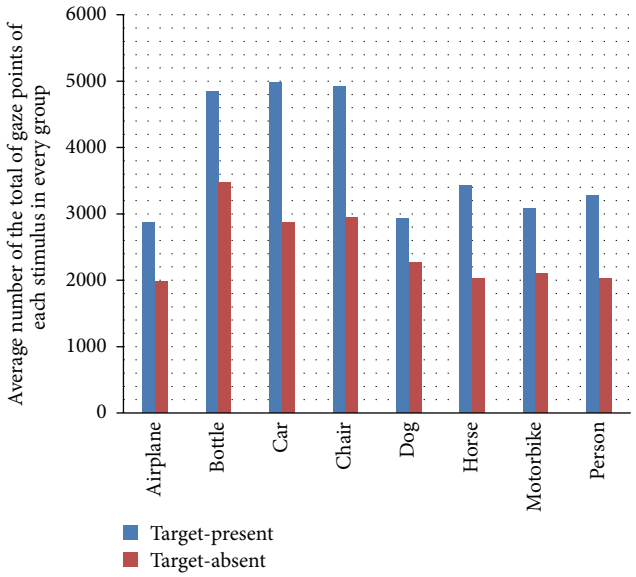


Figure 3: Average number of the total of gaze points of each stimulus in every group.

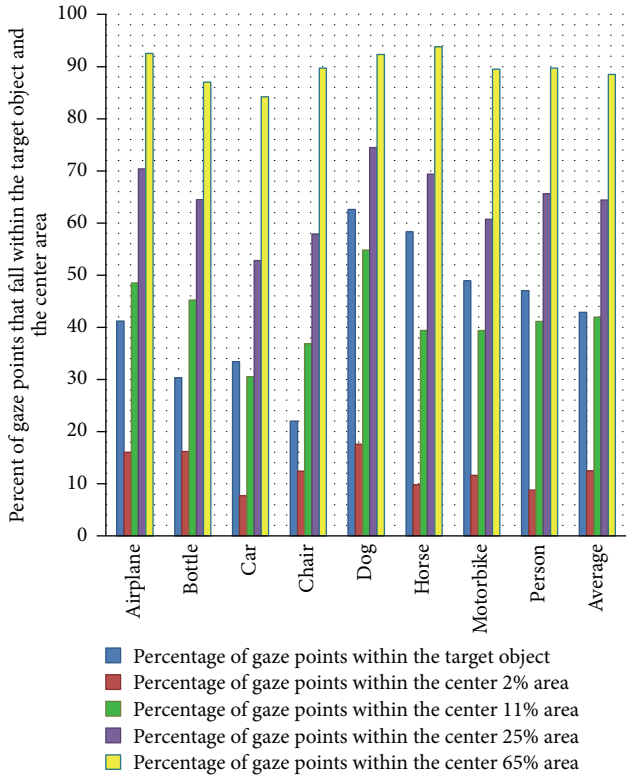


Figure 4: Percent of gaze points that fall within the target object and the center area.

and 65% of the image. Figure 4 summarizes the results. First of all in two cases, the percentages both are above chance level. The differences seen in Figure 4 are statistically significant: the center 25% of the image better attracts human fixations than the target object area. This effect was mostly driven by subject's sidelong glance, for which human fixations are always around target object. But even so, the graphs in Figure 4 clearly indicate that the location of target object (the center area) and the area of target object will attract human fixations.

3.2.5. Objects of Interest. According to Judd et al. [12, 42], if stimuli have one or more humans, gaze points should mainly locate on the human faces. However, in our test, this situation is not similar.

Figure 5 shows heat map of stimuli in which have one or more humans. From Figure 5, we can know the following:

(a) For one stimulus, it has different heat map in different situation.

(b) If the human is the target object, many gaze points still locate on the human face.

(c) When subjects search target in the stimuli, they can ignore the other objects and pay all attention to the target object.

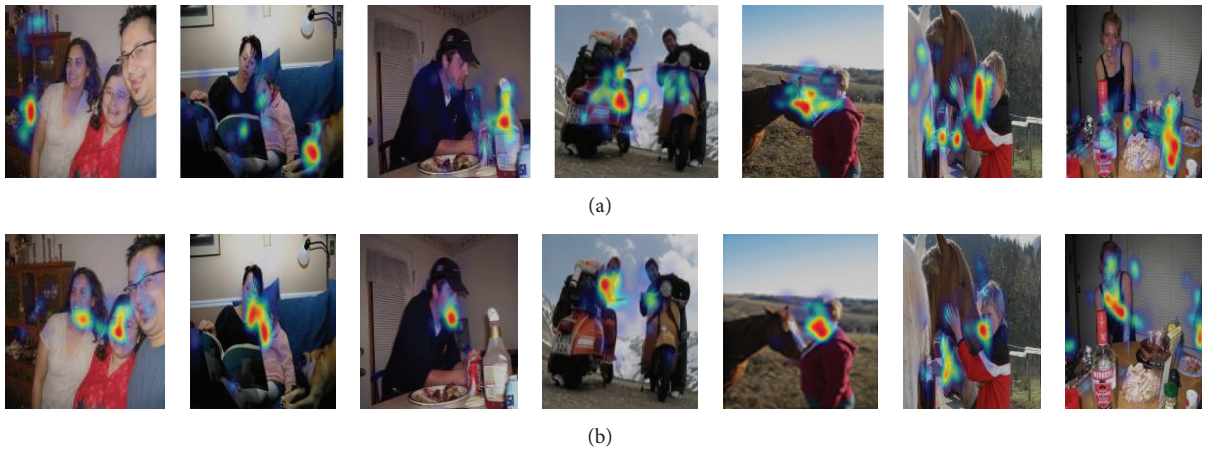
From what we have discussed above, we know that in our test whether some object is of interest depends on the task.

4. Learning-Based Saliency Model

In contrast to previous computational models that combine a lot of biologically plausible filters together to estimate visual saliency, we use a learning approach to train a classifier directly from human eye tracking data. For each image, we precomputed the feature maps for every pixel of the image resized to 200 × 200 and used the maps to train our model. Figure 6 shows the feature maps. Through analyzing our dataset, we promoted low-level, high-level, and center prior features.

Low-level features, intensity, orientation, and color contrast have long been seen as significant features for bottom-up saliency. We include the three channels corresponding to these image features as calculated by Itti and Koch's saliency method [43]. Regarding high-level features, according to our data analysis, we found that humans gaze points always located on target object. So we used the location of target object as the high-level features. Firstly, bounding boxes around objects were labeled and we used them as the target object's area. Secondly, in the boxes, we used the distance of every pixel to the center of box instead of the pixel. Finally, out of boxes, we used zero instead of the pixel. Center bias, when humans take pictures, they naturally frame an object of interest near the center of the image. For this reason, we include a feature which indicates the distance to center of each pixel [12].

To evaluate our model, we followed the 5-fold cross validation method. The method partitions the database into five subsets randomly, each with M images. Every subset is selected sequentially as a test set and the remainders serve as the training set. Each time we trained the model from 4 parts and tested it over the remaining part. Results are then averaged over all partitions. From the ground truth gaze point



(a)

(b)

Figure 5: The figure shows the heat map of stimuli. (a) It shows the target-present's heat map but human is not target object. (b) It shows the target-present's heat map but human is target object.

Table 2: The table shows the average (Avg) and the corresponding standard deviations (STD) of the weight of attribute in each category. For every category, the bold weight is the first and the second is italic weight.

Category	Color		Intensity		Orientation		Target		Center bias	
	Avg	STD	Avg	STD	Avg	STD	Avg	STD	Avg	STD
Airplane	0.0319	0.00005	−0.0154	0.00002	0.0098	0.00002	*0.1201*	0.00012	**−0.4344**	0.00025
Bottle	0.0346	0.00006	0.0424	0.00006	0.0294	0.00004	*0.1206*	0.00009	**−0.3586**	0.00019
Car	0.0073	0.00001	0.0112	0.00002	−0.0159	0.00002	**0.2575**	0.00016	*−0.2418*	0.00012
Chair	0.0234	0.00003	0.0578	0.00006	0.1002	0.00011	**0.2766**	0.00013	*−0.1348*	0.00008
Dog	0.0066	0.00001	0.0075	0.00001	0.0848	0.00006	*0.1065*	0.00008	**−0.4556**	0.00024
Horse	0.0241	0.00004	−0.0004	0.00000	0.0240	0.00003	*0.1445*	0.00011	**−0.3182**	0.00031
Motorbike	−0.0088	0.00002	0.0166	0.00002	0.0276	0.00004	*0.2001*	0.00015	**−0.2733**	0.00025
Person	−0.0131	0.00003	−0.0291	0.00003	0.0638	0.00006	*0.1241*	0.00007	**−0.3159**	0.00027

map of each image, 20 pixels were randomly sampled from the top 20% salient locations, and 20 pixels were sampled from the bottom 70% salient locations to yield a training set of 3200 positive samples and 3200 negative samples. The purpose of choosing a 1 : 1 sampling ratio is to balance the distributions of positive and negative sample pixels in the same image. We chose samples from top 20% and bottom 70% in order to have samples that were strongly positive and strongly negative. The training samples were normalized to have zero mean and unit variance. The same parameters were used to normalize the test set.

We used the linear support vector machine [44] to train the model which was first used to learn the weight of each low-level, high-level, and center prior attribute in determining the significance in attention allocation. We used models with linear kernels because they are faster to compute, and the resulting weights of attributes are intuitive to understand. For each group, the average (Avg) and the corresponding standard deviations (STD) across the number of experiment executions of the learned weight of each attribute are shown in Table 2. It is clear that the attribute of center bias and the location of target object have the higher weight than others. Obviously, in the dog group, the weight of center bias is stronger than others. However, in the chair group, the weight of the location target object is stronger than others. For this phenomenon, the flowing may be critical. The areas of target object may contribute to the phenomenon. But we do not know the detailed relations. The weight of attribute also agrees with previous finding in figure-ground perception that, during visual search tasks, in which subjects are asked to find a particular target in a display, top-down processes play a dominant role in the guidance of eye movements.

5. Evaluation

To measure performance of saliency models, we performed comparisons of our models with the MIT model [3] which is one of the best models in predicting context-free human gaze. The model incorporated bottom-up saliency and high-level image semantics and works well in predicting saliency in a free-viewing context. To make the result comparable, the MIT model is trained on the same training set as our method. Figure 7 shows heat maps of our model and the compared model. This is result for one image in each group. We conducted our experiment on 160 images randomly selected.

Figure 8 shows Receiver Operating Characteristic (ROC) for our model and MIT model. These curves show the

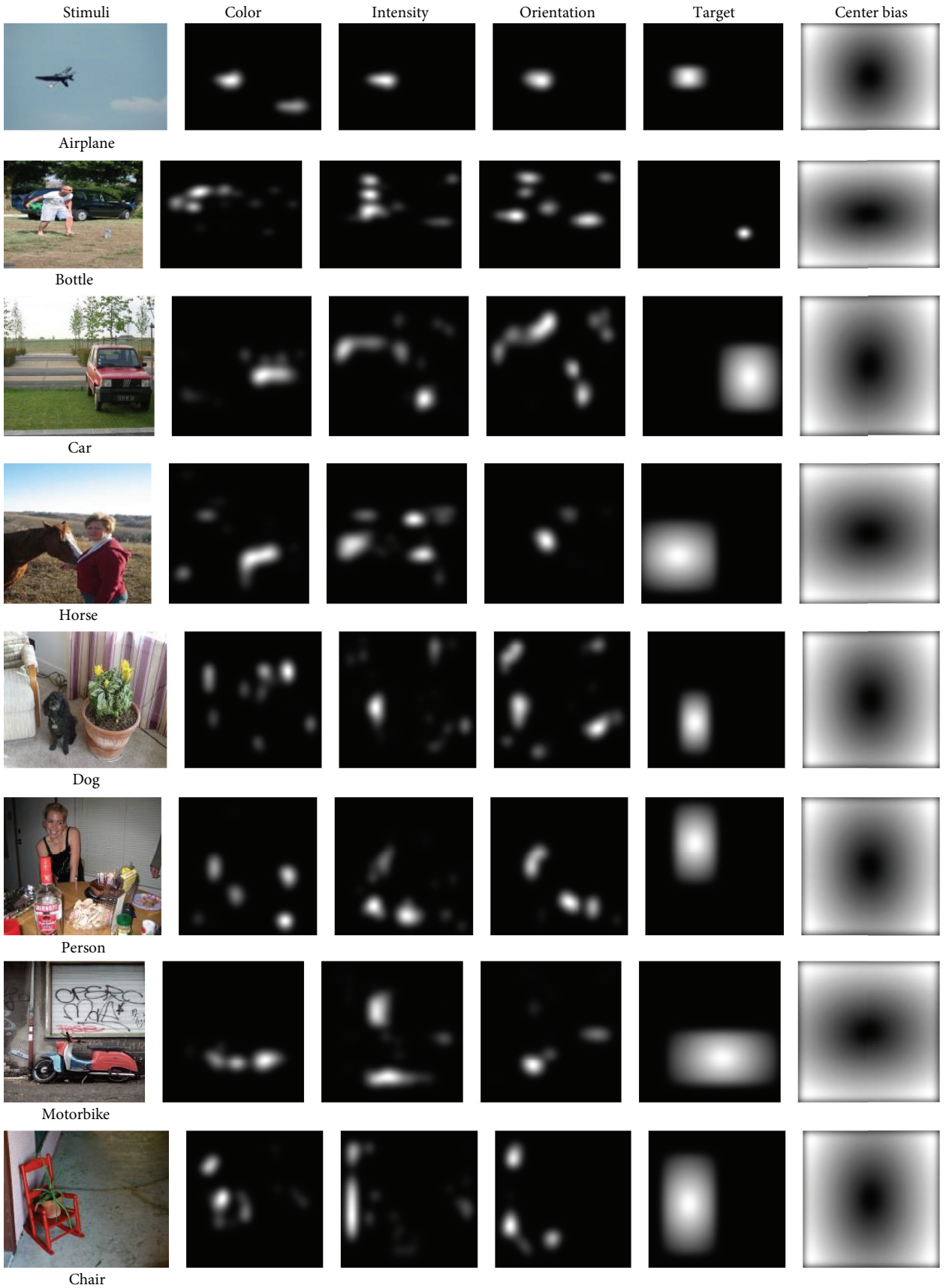


Figure 6: The figure shows the low-level feature maps such as color, intensity, orientation, and high-level feature maps such as the location of target object, finally, center-bias feature map.

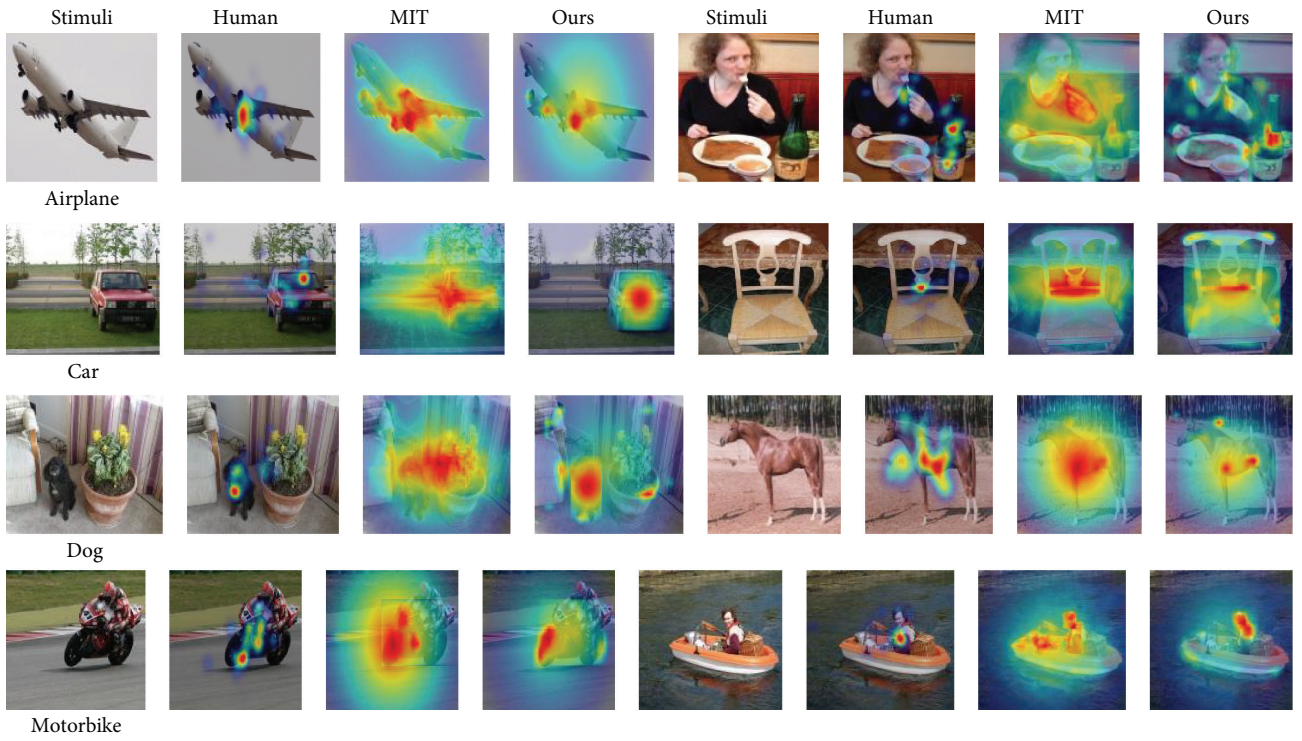


Figure 7: The figure shows the heat maps, which are generated by our model and MIT model. They were trained by the same gaze points and used the same training method.

Table 3: The table shows the average (Avg) and the corresponding standard deviations (STD) of the AUC in each category.

Model	Category							
	Airplane	Bottle	Car	Chair	Dog	Horse	Motorbike	Person
MIT								
Avg	0.8572	0.7881	0.7865	0.8152	0.8639	0.8583	0.8563	0.7962
STD	0.0016	0.0012	0.001	0.0015	0.0006	0.0008	0.0013	0.0011
Ours								
Avg	0.8635	0.8566	0.8873	0.9015	0.8665	0.8663	0.8563	0.8893
STD	0.0012	0.0006	0.0005	0.0004	0.0006	0.0009	0.0007	0.0007

proportion of gaze points that fall within the saliency map predicated by saliency model (detection rate) in relation to the proportion of the image area selected by the saliency map (false alarm rate). Our saliency models were generated by a weighted linear combination of the feature maps using the learned weights of each attribute. It shows how well the gaze points of each subject can be predicted by saliency model. For each category, we calculate the average (Avg) and the corresponding standard deviations (STD) across the number of experiment executions of the area under the ROC curve (AUC), which is shown in Table 3.

It can be seen that, for the MIT model, the performance is not always well; however, our model is better than MIT. For example, in bottle, car, and chair category, MIT model predicted gaze points regions with lower accuracy (AUC = 0.7881, AUC = 0.7865, and AUC = 0.8152) than our models (AUC = 0.8566, AUC = 0.8873, and AUC = 0.9015). From Table 3, we know that the weight of location of target object is first in the car and chair category. So, the promotion of accuracy mainly results from target guidance factor. However, even our model could not compete with human agreement.

6. Discussions and Conclusions

According to Figure 8 and Table 3, it is obviously shown that, for the bottle, car, and chair category, MIT model has lower performance, while our model has larger better performance than it. The main factor is that in these categories target object is small or not salient, so when subjects are free-viewing, they are not saliency map. However, in the task-oriented attention, they become the saliency map; that is why free-viewing model is not appropriately task-oriented.

As we all know several recent datasets [10–12, 45] all set the free-viewing time to 2–5 s per image. In our paradigm, the time was given to the subjects, which is mostly motivated by the following factors. If the viewing duration is too short,

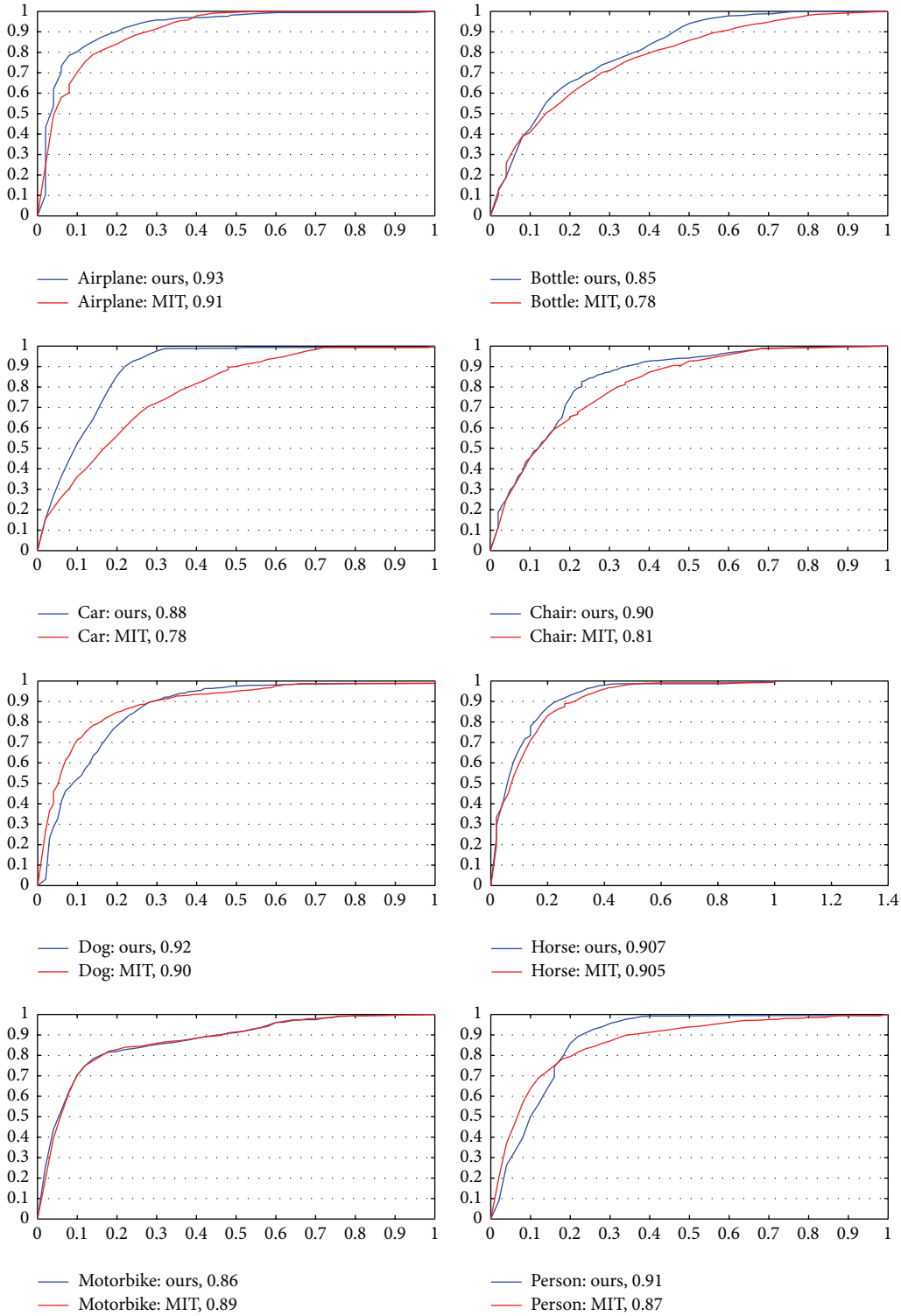


FIGURE 8: The figure shows the Receiver Operating Characteristic (ROC) for our model and MIT model. For each picture, the false alarm rate, on the x-axis, and the detection rate, on the y-axis. Besides, for each category, we calculate the averaging AUC scores of all the predictions, which are shown in above picture.

subjects might not have enough time to find the target objects and also promote the weight of center bias. On the other hand, if the viewing duration is too long, as the viewing proceeded, top-down or other factors (e.g., subjects feel bored and tired) come into play and gaze points become noisier. In addition, if the viewing duration is too long, gaze points may become the free-viewing.

Daily human activities involve a preponderance of visually guided actions, requiring observers to determine the presence and location of particular objects. Based on it, we researched how consistent human gaze points are across an image. Previous research and experience have shown that the gaze point location of several humans is strongly indicative of where a new subject will look, whether target-absent, and target-present. We implemented computational model for target-present in visual search and evaluated how well the model predicted subject's gaze points locations. In our experience, when subjects looked at a scene with a particular task, they consistently payed greater attention to the location of target objects and ignored the other saliency objects, such as text and people. So, our model combined the location of target as the high-level features. Ultimately, the model of attentional guidance predicted 95% of human agreement with the location of target object component providing the most explanatory power.

In this work we make the following contributions. We develop a collection of eye tracking data from the 11 people across 1307 images and have made it public for research use. It is the largest eye tracking database based on the visual search, which provides not only the accurate subjects' gaze points but also segmentation of target object for each image. In this search task, the location of target object is a dominating factor. We use machine learning to train a bottom-up, top-down model of saliency based on low-level, high-level, and center prior features. Finally, to demonstrate performance of our model, the same method was used to train MIT model.

For future work we may be interested in researching that the subjects' gaze points are tightly clustered in very small and specific regions, but our model selects a much more general region containing many objects without gaze points. We believe that the features of target object such as size, scale, and shape will lead subjects to fixate on target, which should be researched more carefully.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The work is supported by NSF of China (nos. 61117115 and 61201319), the Fundamental Research Funds for the Central Universities, and NWPU "Soaring Star" and "New Talent and Direction" Program.

References

[1] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognitionwith visual attention," http://arxiv.org/abs/1412.7755.

[2] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[3] A. D. Hwang, H.-C. Wang, and M. Pomplun, "Semantic guidance of eye movements in real-world scenes," *Vision Research*, vol. 51, no. 10, pp. 1192–1205, 2011.

[4] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 185–198, 2010.

[5] W. S. Geisler and J. S. Perry, "Real-time foveated multiresolution system for low-bandwidth video communication," in *Human Vision and Electronic Imaging III*, vol. 3299 of *Proceedings of SPIE*, pp. 294–305, 1998.

[6] K. Shubina and J. K. Tsotsos, "Visual search for an object in a 3D environment using a mobile robot," *Computer Vision & Image Understanding*, vol. 114, no. 5, pp. 535–547, 2010.

[7] C. Siagian and L. Itti, "Biologically inspired mobile robot vision localization," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 861–873, 2009.

[8] M. Casares, S. Velipasalar, and A. Pinto, "Light-weight salient foreground detection for embedded smart cameras," *Computer Vision & Image Understanding*, vol. 114, no. 11, pp. 1223–1237, 2010.

[9] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

[10] M. Cerf, J. Harel, W. Einhäuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," *Neural Information Processing Systems*, vol. 20, pp. 241–248, 2007.

[11] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T.-S. Chua, "An eye fixation database for saliency detection in images," in *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 30–43, Springer, Berlin, Germany, 2010.

[12] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV '09)*, pp. 2106–2113, Kyoto, Japan, September 2009.

[13] I. Van Der Linde, U. Rajashekar, A. C. Bovik, and L. K. Cormack, "DOVES: a database of visual eye movements," *Spatial Vision*, vol. 22, no. 2, pp. 161–177, 2009.

[14] J. H. Van Hateren and A. Van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings of the Royal Society B: Biological Sciences*, vol. 265, no. 1394, pp. 359–366, 1998.

[15] K. A. Ehinger, B. Hidalgo-Sotelo, A. Torralba, and A. Oliva, "Modelling search for people in 900 scenes: a combined source model of eye guidance," *Visual Cognition*, vol. 17, no. 6-7, pp. 945–978, 2009.

[16] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, no. 4, pp. 219–227, 1985.

[17] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[18] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau, "A coherent computational approach to model bottom-up visual attention,"

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 802–817, 2006.

[19] D. Reisfeld, H. Wolfson, and Y. Yeshurun, "Context-free attentional operators: the generalized symmetry transform," *International Journal of Computer Vision*, vol. 14, no. 2, pp. 119–130, 1995.

[20] G. Heidemann, "Focus-of-attention from local color symmetries," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 26, no. 7, pp. 817–830, 2004.

[21] G. Kootstra, A. Nederveen, and B. de Boer, "Paying attention to symmetry," in *Proceedings of the British Machine Vision Conference*, pp. 1115–1125, Leeds, UK, September 2008.

[22] E. Erdem and A. Erdem, "Visual saliency estimation by nonlinearly integrating features using region covariances," *Journal of Vision*, vol. 13, no. 4, article 11, 2013.

[23] S. Marat, T. Ho Phuoc, L. Granjon, N. Guyader, D. Pellerin, and A. Guérin-Dugué, "Modelling spatio-temporal saliency to predict gaze direction for short videos," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 231–243, 2009.

[24] N. Murray, M. Vanrell, X. Otazu, and C. A. Parraga, "Saliency estimation using a non-parametric low-level vision model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 433–440, IEEE, Providence, RI, USA, June 2011.

[25] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vision Research*, vol. 49, no. 10, pp. 1295–1306, 2009.

[26] L. Zhang, M. H. Tong, and G. W. Cottrell, "SUNDAy: saliency using natural statistics for dynamic analysis of scenes," in *Proceedings of the 31st Annual Cognitive Science Conference*, Amsterdam, Netherlands, 2009.

[27] N. D. B. Bruce and J. K. Tsotsos, "Spatiotemporal saliency: towards a hierarchical representation of visual saliency," in *Attention in Cognitive Systems*, vol. 5395, pp. 98–111, Springer, Berlin, Germany, 2009.

[28] T. Avraham and M. Lindenbaum, "Esaliency (extended saliency): meaningful attention using stochastic image modeling," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 4, pp. 693–708, 2010.

[29] B. Schölkopf, J. Platt, and T. Hofmann, "Graph-based visual saliency," *Advances in Neural Information Processing Systems*, vol. 19, no. 2006, pp. 545–552, 2006.

[30] H. J. Seo and P. Milanfar, "Nonparametric bottom-up saliency detection by self-resemblance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 45–52, Miami, Fla, USA, June 2009.

[31] T. Liu, Z. Yuan, J. Sun et al., "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[32] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, IEEE, Minneapolis, Minn, USA, June 2007.

[33] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search," *Psychological Review*, vol. 113, no. 4, pp. 766–786, 2006.

[34] M. Cerf, E. P. Frady, and C. Koch, "Faces and text attract gaze independent of the task: experimental data and computer model," *Journal of Vision*, vol. 9, no. 12, pp. 74–76, 2009.

[35] Q. Zhao and C. Koch, "Learning a saliency map using fixated locations in natural scenes," *Journal of Vision*, vol. 11, no. 3, article 9, pp. 1–15, 2011.

[36] A. Borji, M. N. Ahmadabadi, and B. N. Araabi, "Cost-sensitive learning of top-down modulation for attentional control," *Machine Vision and Applications*, vol. 22, no. 1, pp. 61–76, 2011.

[37] R. J. Peters and L. Itti, "Beyond bottom-up: incorporating task-dependent influences into a computational model of spatial attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, Minneapolis, Minn, USA, June 2007.

[38] F. Baluch and L. Itti, "Mechanisms of top-down attention," *Trends in Neuroscience*, vol. 34, no. 4, pp. 210–224, 2011.

[39] M. Pomplun, "Saccadic selectivity in complex visual search displays," *Vision Research*, vol. 46, no. 12, pp. 1886–1900, 2006.

[40] J. Zelinsky Gregory, W. Zhang, B. Yu, X. Chen, and D. Samaras, "The role of top-down and bottom-up processes in guiding eye movements during visual search," in *Proceedings of the 19th Annual Conference on Neural Information Processing Systems (NIPS '05)*, vol. 18 of *Advances in Neural Information Processing Systems*, pp. 1569–1576, MIT Press, Cambridge, Mass, USA, 2005.

[41] W. Jian and Z. Xinbo, "Analysis of eye gaze points based on visual search," in *Proceedings of the IEEE International Conference on Orange Technologies (ICOT '14)*, pp. 13–16, Xian, China, September 2014.

[42] M. Jiang, J. Xu, and Q. Zhao, "Saliency in crowd," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8695 of *Lecture Notes in Computer Science*, pp. 17–32, 2014.

[43] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, no. 10–12, pp. 1489–1506, 2000.

[44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: a library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. 12, pp. 1871–1874, 2008.

[45] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Advances in Neural Information Processing Systems 18.3*, pp. 155–162, MIT Press, 2005.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 8343842, 17 pages
http://dx.doi.org/10.1155/2016/8343842

Research Article

Characterization of Visual Scanning Patterns in Air Traffic Control

Sarah N. McClung[1] and Ziho Kang[2]

[1]*School of Electrical and Computer Engineering, University of Oklahoma, 110 W. Boyd Street, Devon Energy Hall 150, Norman, OK 73019-1102, USA*
[2]*School of Industrial and Systems Engineering, University of Oklahoma, 202 West Boyd Street, No. 116, Norman, OK 73019, USA*

Correspondence should be addressed to Ziho Kang; zihokang@ou.edu

Received 27 November 2015; Accepted 14 January 2016

Academic Editor: Francesco Camastra

Characterization of air traffic controllers' (ATCs') visual scanning strategies is a challenging issue due to the dynamic movement of multiple aircraft and increasing complexity of scanpaths (order of eye fixations and saccades) over time. Additionally, terminologies and methods are lacking to accurately characterize the eye tracking data into simplified visual scanning strategies linguistically expressed by ATCs. As an intermediate step to automate the characterization classification process, we (1) defined and developed new concepts to systematically filter complex visual scanpaths into simpler and more manageable forms and (2) developed procedures to map visual scanpaths with linguistic inputs to reduce the human judgement bias during interrater agreement. The developed concepts and procedures were applied to investigating the visual scanpaths of expert ATCs using scenarios with different aircraft congestion levels. Furthermore, oculomotor trends were analyzed to identify the influence of aircraft congestion on scan time and number of comparisons among aircraft. The findings show that (1) the scanpaths filtered at the highest intensity led to more consistent mapping with the ATCs' linguistic inputs, (2) the pattern classification occurrences differed between scenarios, and (3) increasing aircraft congestion caused increased scan times and aircraft pairwise comparisons. The results provide a foundation for better characterizing complex scanpaths in a dynamic task and automating the analysis process.

1. Introduction

Air traffic controllers are considered to have a highly stressful occupation due to the weight of their responsibilities and the constant expectation of their faultless performance [1, 2]. They monitor aircraft, communicate with pilots, and solve conflicts that threaten either loss of separation (LOS) of a minimum allowed distance between aircraft or wake turbulence [3]. Since 1980, industrial air traffic has averaged over 5% growth each year [4] and continues to steadily rise [5], causing ATCs to experience more difficulty with their tasks [6, 7]. Each ATC is assigned a sector in space and as the aircraft traffic increases, the sectors become more crowded [6]. Overload and scenario difficulty has been shown to decrease ATC performance [8], and since more aircraft cause higher probability of errors and ATCs are required to make no mistakes, it is highly important to aid ATCs by providing insight to efficient training methods and utilizing as much automation as possible.

Previous research verified that one way to develop efficient training programs is by allowing novices to view expert ATC visual scanpaths to teach the novices the highest performing scanning strategies at the quickest rate [9, 10]. This method was appropriate because monitoring eye movements can aid in understanding user intent [11]. A scanpath is a sequential eye movement across a display [12–14], as depicted in Figure 1. The points represent eye fixations and the lines represent saccades, which are movements between fixation points [15]; the scanpath moves sequentially from point 1 to point 8. Eye tracking devices, such as those under monitors, have been used to successfully record, observe, and analyze visual scanpaths to improve computer interfaces [16, 17]. Because research has shown that understanding ATC cognitive processes is useful [18, 19] and observing scanpaths is a

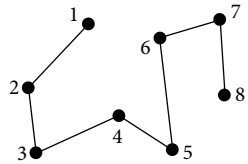


Figure 1: Scanpath example.

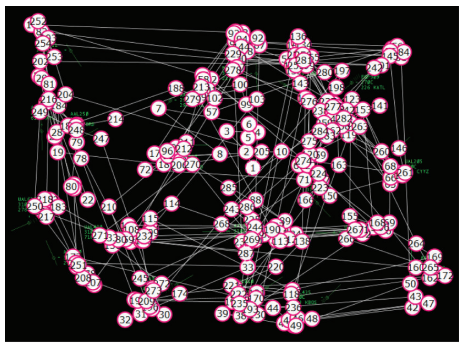


Figure 2: Example of a real scanpath overlaid on a static display of 1.5-minute duration.

suitable method [20], automating detailed characterization of scanpaths would also be valuable to provide effective training techniques. If scanpath characterization can be accurately automated, expert ATC scanpaths can be collected, recorded, and characterized for novices to watch for deeper understanding. Additionally, novices can receive more feedback on their own scanpaths while running simulations to test the efficiency of their strategies as well as their performance.

However, there is a lot of uncertainty when attempting to characterize scanpaths. According to ATC linguistic inputs, scanning strategies can be conceptually described as being circular, linear, trajectory, regional, augmented, proximity-based, or density-based [9, 21], but realistically identifying a scanpath into only one of the listed categories is unlikely due to overlap caused by their elementary definitions. Therefore, the challenge exists in attempting to correctly map the ATC self-reported strategies, or patterns, to each of their actual scanpaths. This is difficult because scanpaths grow in complexity with time and become highly difficult to classify.

Figure 2 shows a scanpath of 1.5-minute duration in which correct classification into the strategies provided by the ATCs would be unlikely. In addition to finding multiple overlapping patterns, the patterns can be incomplete, chaotic, or ambiguous.

Another major concern that causes classification uncertainty is that some scanpaths appear to cause a pattern that was not intended. Most expert ATCs intend to use a particular strategy, but extracting exactly what they intended from the data can be difficult. For example, an ATC can intend to use a circular scanpath, but it cannot be identified by the eye tracking data alone because of constant back-and-forth comparisons between aircraft and lack of a complete circle. The circle can switch directions several times from clockwise to counterclockwise in semicircles and can be interrupted by comparisons that cause the ATC to look across the screen in linear motions. When observing the plotted data, it is possible that the intended circular scanpath can be classified instead as mixed between linear and augmented. To visualize this issue, Figure 3 demonstrates additional fictional examples of basic scanpaths with their corresponding shapes. The numbers show the order the aircraft were viewed in, the thick red lines show the saccades, and simplified representations are below in blue. Choosing one pattern to label (a) and (b) from the many options available unfortunately relies on a judgement call, especially since the patterns are not exclusive. Scanpath (a) can be argued to be circular from fixations 3 to 10, linear from 1 to 8 or 5 to 12, or augmented from 1 to 12 moving from quadrants Q2, Q4, Q1, and then Q3. Scanpath (b) can be linear, trajectory, or density-based from fixations 1 to 9. Similar issues frequently occur when viewing ATC scanpaths. The patterns require thresholds and possibly hierarchical order to determine which strategy was dominant.

Other issues to consider are the level of influence on a scan from the number of aircraft, the difficulty of the scenario, and the spatial layout of the aircraft. Overall, the type of scan depends on the ATC's strategy and level of influence from the mentioned variables which often leads to multiple incomplete, local, or unclear patterns that prove highly challenging to characterize. For example, in [9], circular is defined as observing circular motions in the scanpath; however the details describing circular motions were not defined. Therefore, it is possible to have human judgement bias during inter-rater agreement if specific procedures to characterize and classify the scanpaths are not developed.

In order to begin addressing these problems in a simplified manner, consider limiting the scanpath patterns to only being circular, linear, and mixed as provided in Table 1. In this research, only circular and linear patterns were searched for because they are the simplest strategies to witness and most frequently used by expert ATCs [21] and depend only on scanpath shape. Ideal examples include counterclockwise, clockwise, and spiral for circular patterns and horizontal, vertical, and diagonal for linear patterns. The realistic examples are representations of the ideal examples; the mixed pattern uses both circular and linear movements to complete the scan, so there is no ideal form. The realistic examples are fictional, overlaid on a low congestion scenario of 12 aircraft to conceptually demonstrate the categories. Note that patterns can have a combination of their characteristics such as the second circular and linear examples that change direction before completion.

Due to the different possible interpretations and the high magnitude of complexity, scanpaths are difficult to classify into the patterns provided by the expert ATCs. Although some ideal and realistic examples are provided, there is still a challenge in mapping each scanpath to those selected patterns. Patterns are described conceptually, but there are no given thresholds or mathematical representations to confidently identify them. For example, how is a linear movement mathematically defined from an ATC verbal description (such as moving left to right while zigzagging [21])? Moreover, if linear movements are successfully represented in mathematical form, how are realistic scanpaths analyzed

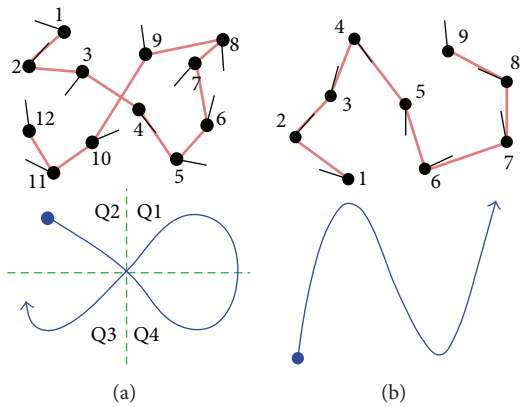
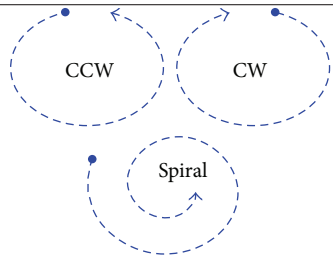
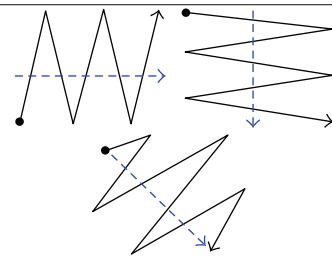
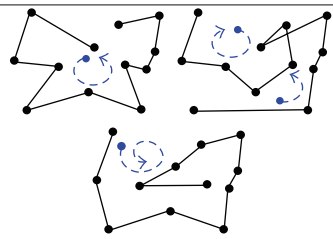
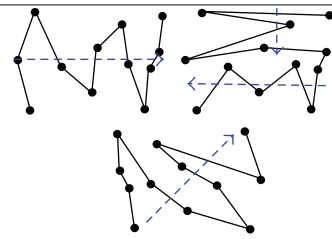
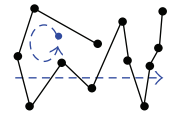


Figure 3: Examples of scanpaths with corresponding simplified patterns. The scanpaths overlaid on aircraft are shown above with saccades in red and the interpreted patterns are shown below in blue. (a) Circular, linear, and regional patterns can be identified. (b) Linear, trajectory, and density-based patterns can be identified.

Table 1: Ideal and realistic examples of relevant scanpath patterns.

	Circular	Linear	Mixed
Ideal examples			N/A
Realistic examples			

given that they do not move in perfect or predictable ways? Many conditions need to be considered, such as the random fluctuations that occur in scans including back-and-forth movements to previous aircraft. Although many algorithms were developed to compare and analyze visual scanpaths [12, 22–32] and their capabilities and limitations are provided in detail in [12], the methods were limited to comparing scanpaths but not mapping the visual scanpaths to strategies verbalized by the experts.

Naturally, it is difficult to develop mathematical models or algorithms based on verbal descriptions provided by the ATCs. The descriptions require more depth; after refining them, it may be possible to divide the scanpaths into patterns that can be confidently classified based on certain criteria. Pattern identification requires thresholds that address points of deviation, the percentage of aircraft viewed following a given pattern, and methods to correctly identify the pattern. Otherwise, many patterns can be claimed to be used in a scan although only one is dominant and intended by the ATC. If terminologies and procedures were predetermined and applied in a systematic manner, then more consistent and in-depth discoveries could be reached. Therefore, the purpose of this paper is to (1) introduce new terminology, (2) apply filtration methods to scanpaths to simplify their representation before judgement, (3) provide procedures that behave as a conceptual framework for raters during pattern classification, and (4) apply the proposed procedures to characterize scanpaths and compare their results across scenarios with different number of aircraft. This work is meant to ease characterization of scanpaths, increase characterization accuracy, and contribute to future automation of scanpath characterization.

2. Proposed Methodology

2.1. Terminology for Identifying Visual Scanning Strategies. New terminology is introduced in order to filter out the complex scanpaths into manageable forms that can lead to

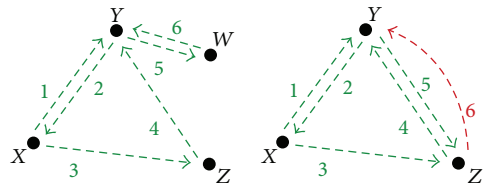


FIGURE 4: Local scan without and with a comparison. The black dots represent aircraft named W, X, Y, and Z. The numbered arrows indicate the order in which the aircraft were observed. Green indicates a local scan and red indicates a comparison. (a) Local scan between W, X, Y, and Z with no comparisons. (b) Local scan between X, Y, and Z and comparison between Y and Z.

representations of the strategies provided by ATCs. Assume there are N number of aircraft present on the radar display for the definitions provided below:

(i) *Raw scanpath*: the raw scanpath is the entire scanpath including all fixations and saccades from the beginning to the end of the task being carried out by the ATC to solve all conflicts in the scenario.

(ii) *Global scan*: a global scan is a complete observation of all N aircraft.

(iii) *Local scan*: a local scan is an observation of a group of aircraft with possible comparisons, where the group contains 2 to $(N - 1)$ aircraft.

(iv) *Comparison*: a comparison occurs when aircraft are consecutively scanned at least twice after already being viewed once before for a total of three or more observations until moving on to different aircraft.

(v) *Initial global scanpath (IGS)*: the IGS is the first complete observation of all N aircraft on the display. It includes all fixations and saccades that occurred until all aircraft were visited.

(vi) *Extracted IGS*: the extracted IGS is the IGS with comparisons removed, or filtered out; it applies the initial filter to the IGS.

(vii) *Fundamental IGS*: the fundamental IGS is the IGS without local scans; it applies the most intense filter and extracts the simplest form of the IGS. It displays the order each aircraft was viewed during the IGS.

During a global scan, if all N aircraft are visited, then the following eye fixation starts the next possible global scan. In a raw scanpath, there can be multiple global scans or none if all the aircraft were never viewed. Similarly, a global scan can include multiple local scans or none, and a local scan can include multiple comparisons or none. Particularly, they exist as subsets of each other as shown below:

$$\text{(Raw Scanpath)} \subseteq \text{(Global Scans)} \subseteq \text{(Local Scans)} \subseteq \text{(Comparisons)}.$$

The concept of local scans and comparisons is derived from the definition of visual groupings [33], or a significant amount of transitions between aircraft. Comparisons are made during conflict resolution between aircraft. When a comparison is finalized by an ATC moving on to different aircraft, the ATC can either return again to the same aircraft to perform another comparison, compare a different group of aircraft, or continue scanning. A comparison is always a local scan, although local scans are not always comparisons if the aircraft are merely scanned without repetition. Figure 4 illustrates the difference between a local scan and comparison; (a) shows a local scan between W, X, Y, and Z with no comparisons while (b) shows a local scan between X, Y, and Z with a comparison beginning during the sixth movement between Y and Z shown in red. Note that the aircraft group shown (W, X, Y, Z) are only 4 out of many more aircraft on a radar display; thus viewing them does not complete a global scan.

Only the IGS was analyzed for number of comparisons and scanpath pattern because there are less chances of local scans and comparisons during that time. After the IGS is completed, all the aircraft have been viewed and the remaining time is most likely spent on conflict resolution which does not require additional global scans. Individual ATC scanning strategies are most likely to be witnessed during the IGS with more ease and clarity compared to the rest of the scan.

However, the problem remains that the IGS usually consists of multiple local scans with comparisons causing it to still be difficult to classify. Applying filters to the repetitive movements eases the classification process; therefore the extracted and fundamental IGSs were used. The extracted IGS represents how raters should naturally observe scanpaths while watching and judging scanning strategies; it disregards comparisons between aircraft but has to consider local scans. The fundamental IGS is the most simplified representation of a scanpath; as previously mentioned, it disregards local scans which implies excluding comparisons as well. It has N number of fixations and $(N - 1)$ number of connections between fixations. The connections between the fixations are not necessarily saccades since they merely show the path to the next viewed aircraft by the ATC without considering any repeat observations to previous aircraft. Particularly, the aircraft are numbered from 1 to N in the order they were fixated on; then connections are drawn sequentially between the fixations.

Once the extracted and fundamental IGSs are obtained, their patterns can be classified. The scanpath strategies consist of seven known pattern categories: circular, linear, trajectory, regional, augmented, density-based, and proximity-based categories [9, 21]. Two additional categories that allow all scanpaths to be identified are "mixed" and "other." The most popular strategies used by expert ATCs are circular and then linear [9, 21], and as previously explained, they are also shape

INPUT IGS
OUTPUT The number of comparisons made during the IGS
Step 1. Define 3 states as *Scanned*, *Potential Comparison*, and *Comparison*.
Step 2. If an aircraft is viewed for the first time, it is placed in *Scanned*.
Step 3. If an aircraft from *Scanned* is viewed for the second time, it is copied into *Potential Comparison*.
Step 4. If the next aircraft viewed is also from *Scanned*, it is copied into *Potential Comparison* as well. Otherwise, if a new aircraft is viewed, it is placed in *Scanned* and the *Potential Comparison* state clears.
Step 5. If *Potential Comparison* is occupied with aircraft and one of those aircraft are viewed again, it is copied into *Comparison*.
Step 6. As long as aircraft in *Potential Comparison* continue to be viewed, they are copied into *Comparison* as one comparison.
Step 7. When an aircraft is viewed that is not in *Potential Comparison*, the comparison is complete, the number of comparisons (N_{Comp}) is incremented, and the states *Potential Comparison* and *Comparison* are cleared. The aircraft that broke the comparison is either new and belongs in *Scanned* or is already in *Scanned* and is copied into *Potential Comparison* to possibly begin the next comparison.

Procedure 1: Counting the number of comparisons.

INPUT Fundamental IGS
OUTPUT Circular scanpath
Let the center of the screen be the origin.
Let 0° be assigned in reference to the first eye fixated target (ϑ_1).
Let i be an indicator for sequential eye fixations where $i = \{1, 2, 3, \ldots, k\}$.
Let n be a counter variable where $n = \{1, 2, 3, \ldots, k\}$.
Step 1. Assign each subsequent eye fixated target a degree value (ϑ_i, where $i = \{2, 3, 4, \ldots, k\}$) in reference to the clockwise direction from ϑ_1.
Step 2. Increment n for each sequential increase in ϑ_i (where $\vartheta_{i+1} \geq \vartheta_i$), then go to *Step 4*.
Step 3. Increment n if the next value is ϑ_k, then continue to increment n for each sequential decrease in ϑ_i (where $\vartheta_{i-1} \leq \vartheta_i$), then go to *Step 4*.
Step 4. If $n \geq N/2$, then the pattern is labeled circular.

Procedure 2: Identifying circular scanpaths.

dependent and are the easiest to identify. Thus, for purpose of simplification, this work divides all scanpath patterns into circular, linear, mixed, and other categories which are briefly defined below and previously illustrated in Table 1. Trajectory scans are not included because they are utilized by novices [9], and this research studies expert ATC behavior:

(i) *Circular*: circular scanpaths rotate in a clockwise or counterclockwise motion and include spirals and rectangles. They move along adjacent edges of the screen and tend to end adjacent from where they began.

(ii) *Linear*: linear scanpaths are directional from one side to its opposite and move in zigzags perpendicular to their horizontal, vertical, or diagonal direction. They tend to end opposite from where they began.

(iii) *Mixed*: mixed scanpaths occur when both circular and linear scanpaths occur and can include overlap.

(iv) *Other*: other scans lack confident identification and therefore include patterns that are unknown or categories too difficult to confidently identify including regional, augmented, density-based, and proximity-based scanpaths.

2.2. Characterization Procedure. Based on the definitions in Section 2.1, procedures were developed for identifying the number of comparisons in an IGS and for pattern classification of extracted and fundamental IGSs. The analysis process included counting the number of comparisons in the IGS, then simplifying the IGS to extracted and fundamental forms, and classifying those simplified scanpaths as circular, linear, mixed, or other as indicated in Figure 5.

Procedures were followed to determine the number of comparisons and the type of scanpath. The procedures are meant to be used on IGSs; Procedure 1 analyzes the IGS, and Procedures 2–4 analyze the fundamental IGS. Procedures 2–4 only serve as guidelines for extracted IGSs due to the many fluctuations caused by considering local scans; interrater agreement is still the dominant classification method for extracted IGSs. For fundamental IGSs, the procedures are followed for initial classification of scanpaths, and then interrater agreement is utilized to reassign pattern classification to exceptional cases that are judged incorrectly classified by the procedures.

Procedure 1 was used to count the number of comparisons made during the IGS. When aircraft were viewed for the second time, it was not counted as a comparison because it was possible that the ATC forgot a piece of information or needed to make a confirmation. When the aircraft were immediately

INPUT Fundamental IGS
OUTPUT Linear scanpath
Let (X_i, Y_i) be the coordinates of an eye fixated target where i is an indicator for sequential eye fixations.
Let $X_{\min} = \min\{X_1, X_2, \ldots, X_N\}$, $X_{\max} = \max\{X_1, X_2, \ldots, X_N\}$, $Y_{\min} = \min\{Y_1, Y_2, \ldots, Y_N\}$, and $Y_{\max} = \max\{Y_1, Y_2, \ldots, Y_N\}$.
Step 1. When an eye fixated target is an extreme point on the X or Y axis on the screen, let it be (X_1, Y_1).
Step 2. (*Left-to-right*). If $(X_1, Y_1) = (X_{\min}, Y)$ AND X values increase to $X_{\max}$ AND Y values switch directions between increasing and decreasing at least twice AND $(X_n, Y_n) = (X_{\max}, Y)$, then go to *Step 6*.
Step 3. (*Right-to-left*). Else if $(X_1, Y_1) = (X_{\max}, Y)$ AND X values decrease to $X_{\min}$ AND Y values switch directions between increasing and decreasing at least twice AND $(X_n, Y_n) = (X_{\min}, Y)$, then go to *Step 6*.
Step 4. (*Bottom-to-top*). Else if $(X_1, Y_1) = (X, Y_{\min})$ AND Y values increase to $Y_{\max}$ AND X values switch directions between increasing and decreasing at least twice AND $(X_n, Y_n) = (X, Y_{\max})$, then go to *Step 6*.
Step 5. (*Top-to-bottom*). Else if $(X_1, Y_1) = (X, Y_{\max})$ AND Y values decrease to $Y_{\min}$ AND X values switch directions between increasing and decreasing at least twice AND $(X_n, Y_n) = (X, Y_{\min})$, then go to *Step 6*.
Step 6. If $n \geq N/2$, then the pattern is labeled linear.
Note: Identification of linear scans with diagonal movements are evaluated by repeating the steps after rotating the display by 45 degrees.

Procedure 3: Identifying linear scanpaths.

INPUT Fundamental IGS
OUTPUT Mixed or other scanpath
Step 1. If the scanpath satisfies Procedure 2, but not Procedure 3, then the pattern keeps its circular classification.
Step 2. If the scanpath satisfies Procedure 3, but not Procedure 2, then the pattern keeps its linear classification.
Step 3. If the scanpath satisfies both Procedures 2 and 3, then the pattern is classified as mixed.
Step 4. Otherwise, the scanpath is classified as other.

Procedure 4: Identifying mixed and other scanpaths.

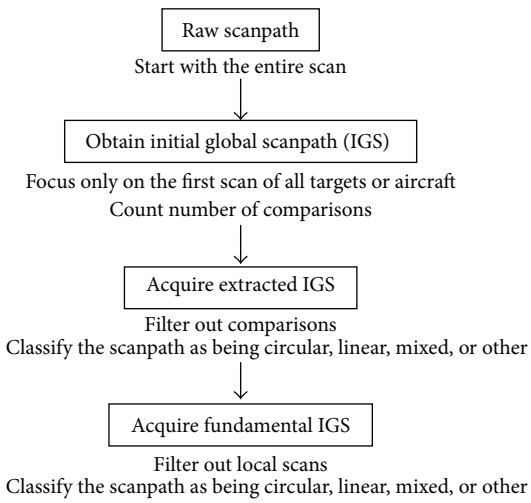


Figure 5: Flow chart for IGS analysis procedure.

observed for a third time, it was assumed that the ATC was making a comparison. In reference to Figure 2, following Procedure 1 leads to no comparisons in (a) and the beginning of the first comparison in (b) although aircraft Y was scanned 3 times in both examples.

Table 2: Application example of Procedure 1.

Step	Scanned	Potential comparison	Comparison	N_{Comp}
1	E, A, B, C	E	—	0
2	E, A, B, C, F	—	—	0
3	E, A, B, C, F	B, E	B	1
4	E, A, B, C, F, D	—	—	1
5	E, A, B, C, F, D	B, E, A	B, A, B, E, A, E	2
6	E, A, B, C, F, D	—	—	2
7	E, A, B, C, F, D	C	—	2

An example is provided that applies Procedure 1 in order to increase clarity. Consider a 20-step sequence between 6 aircraft: A, B, C, D, E, and F. The sequence is as follows:

E→A→B→C→E→F→B→E→B→D→B→E→A→
B→A→B→E→A→E→C.

Table 2 shows how each aircraft appropriately falls into the states introduced in Procedure 1. In the first step, aircraft (E, A, B, C) are viewed for the first time and placed into *Scanned*; then an aircraft (E) is seen again and copied into *Potential Comparison*. In the second step, a new aircraft is viewed (F) and placed into *Scanned*, so the states *Potential Comparison* and *Comparison* clear. In the third step, aircraft from *Scanned*

are viewed again (B, E) and copied into *Potential Comparison*. When an aircraft in *Potential Comparison* is additionally viewed (B), it is copied into *Comparison* where aircraft are allowed to repeat, and N_{Comp} is incremented to 1 (0 + 1 = 1). In the fourth step, a new aircraft is viewed (D) and placed into *Scanned*, and *Potential Comparison* and *Comparison* are cleared. In the fifth step, previous aircraft are viewed (B, E, A) and copied into *Potential Comparison* until they are repeated and therefore copied into *Comparison* (B, A, B, E, A, E) where the comparison continues between those aircraft that are also in *Potential Comparison*, and N_{Comp} is incremented to 2 (1 + 1 = 2). In the sixth step, an aircraft that was not in *Potential Comparison* is viewed, so *Potential Comparison* and *Comparison* are cleared. In the seventh step, since the next aircraft (C) was already in *Scanned*, it is placed in *Potential Comparison*. The number of comparisons incremented until $N_{\text{Comp}} = 2$ for this sequence.

After Procedure 1 is completed for total number of comparisons during the IGS, the extracted and fundamental IGSs are used for Procedures 2–4 for pattern classification. The procedures are general guidelines for classifying extracted IGSs but are closely followed for fundamental IGSs; they are applied by raters; then patterns are determined by interrater agreement. Scans are labeled as being circular or linear when at least 50% of the N total aircraft sequentially follow that pattern, because if a pattern is used on at least half of the aircraft (in the studied scenarios which range from 12 to 20 aircraft on the display), it is most likely not coincidental. Note that the percentage threshold can be adjusted and there is some allowance for points to deviate from the procedures without interrupting the sequential count of aircraft following a pattern; the aircraft count can be paused for 1 or 2 deviation points and then continued for the next aircraft if they continue the pattern. Circular and linear scans depend on the shape made by the scanpath and are independent of ATC intention. If the radar screen is represented by a grid divided into sections, the shape-dependent patterns can be conceptually identified based on the order the aircraft were observed near the outside border of the grid. Hence the center of the grid should be considered a region of error.

Circular patterns are basically achieved when an imaginary bar originating from the center of the display stretching out to the border rotates at least 180°. As the bar rotates, the scanpath must hit the aircraft it touches; circular patterns are made when the scanpath moves to adjacent points along the border, resulting in clockwise or counterclockwise movements as defined in Procedure 2. They usually result in rotating back to the starting point although that is not always the case.

Similarly, linear patterns can be conceptualized by imagining a bar that stretches across the display either vertically, horizontally, or diagonally. As the bar moves from one side to its opposite, the scanpath must hit the aircraft in contact with the bar which results in zigzag motions perpendicular to the direction the bar is moving in. Procedure 3 demonstrates this idea; the scan moves from one side or corner of the grid to the opposite, similar to wave propagation. The scanpath travels perpendicular to the direction of movement in zigzagging motions which creates switching of opposite positions along the border.

Procedures 2–4 provide conceptual frameworks to classify the patterns with chosen thresholds. The classification process is not complete until Procedures 2–4 are applied to each scanpath sequentially. Certain assumptions are made before utilizing Procedures 2 and 3: there is a uniform distribution of aircraft across the display, tolerance is applied by raters to allow some deviation from the ideal patterns, and the procedures are based on assuming that the spatial layout of the multiple targets (or aircraft) are distributed in a uniform manner with random spacing, meaning that they are not equally aligned following a uniform distribution.

The rationale for Procedure 2 is as follows. The angle of the first eye fixated target is always set at $\theta_1 = 0^\circ$. Step 1 assigns θ values to each subsequent eye fixated target in reference to θ_1 (e.g., in Figure 6(a), θ_2 is 15° clockwise from θ_1). Steps 2 and 3 investigate whether there is a consistent increase or decrease of θ values. If increasing, then the visual scan is in a clockwise motion, and if decreasing, then the visual scan is in a counterclockwise motion. Note that for a counterclockwise movement, θ_1 (where $i = 1$) is followed by θ_k (where $i = k$), and then the i values decrease from k. In Step 4, if the identified number of eye fixated targets (n) is over half the total eye fixated targets (N), then we classify the scanpath as circular. Note that the threshold used in Step 6 can be adjusted. A tolerance in sequential increase or decrease must be allowed (e.g., a few θ_i values can increase among an overall decrease of θ_i values) since eye movements are not mechanical and often deviate from ideal mathematical patterns or verbal explanations of individual's search patterns. Examples of circular scanpaths classified by Procedure 2 are shown in Figures 6(a) and 6(c). Each black point represents a target; the first target being fixated on is 0° (θ_1). In Figure 6(a), the remaining points have sequentially increasing θ values which creates a clockwise rotation. In Figure 6(c), the first 10 points apply to a circular pattern, which are over half of the aircraft; therefore it also satisfies Procedure 2.

The rationale for Procedure 3 is as follows. Step 1 defines the starting point of the aircraft fixated on an extreme location on the x- or y-axis of the display. The starting point does not need to be the first eye fixation point on the target since the pattern may not start until a few eye fixations occurred on other targets. Also note that X and Y extrema can include any points relatively near the outside border of the screen to allow some tolerance; they do not have to be the absolute extrema, but they need to be close. Steps 2–5 define the general trend of linear movements of either vertical or horizontal movements. In detail, the movements can be in the overall vertical direction with horizontal zigzags or in the overall horizontal direction with vertical zigzags. The reason that there are 4 differentiated steps is that each step indicates whether the overall direction is increasing from left to right (Step 2), right to left (Step 3), bottom to top (Step 4), or top to bottom (Step 5) and insures they end near the opposite side of the display. In Step 6, if the identified number of eye fixated targets (n) is over half the total eye fixated targets (N), then we classify the scanpath as being linear. Note that the threshold used in Step 6 can be adjusted. Again, a tolerance in the overall

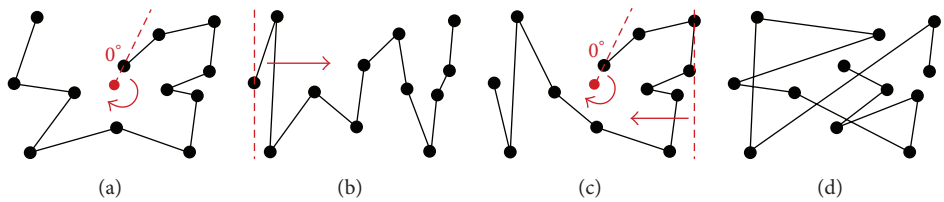


Figure 6: Fundamental scanpath examples that are identified using Procedures 2–4. (a) Circular scanpath identified with Procedures 2 and 4 (Step 1). (b) Linear scanpath identified with Procedures 3 and 4 (Step 2). (c) Mixed scanpath identified with Procedure 4 (Step 3). (d) Another scanpath identified with Procedure 4 (Step 4).

direction must be allowed (e.g., a few X_i (or Y_i) values can increase among overall decrease of X_i (or Y_i) values). Examples of linear scanpaths classified by Procedure 3 are shown in Figures 6(b) and 6(c). The scanpath in Figure 6(b) starts at $X_{\min}$, then the X values consistently increase as the Y values switch directions (between increasing and decreasing) 6 times, and finally it ends at $X_{\max}$; the linear direction is horizontal from left to right. In Figure 6(c), the last 10 points apply towards a horizontal linear pattern from right to left (the third point is excluded, so 9 total points count towards the linear pattern), which include over half of the aircraft; therefore it also satisfies Procedure 3.

The rationale for Procedure 4 is as follows. Steps 1 and 2 cause the scanpath to be analyzed for circular or linear patterns. If the scanpath is exclusively circular or linear, it is classified as such. In Step 3, if the scanpath can satisfy both requirements for circular and linear patterns, it is labeled mixed. Step 4 is provided to assign "other" classification for scanpaths that do not utilize circular or linear patterns at all. After the procedures are followed, interrater agreement is used in order to account for exceptional cases. Utilizing interrater agreement is a necessity due to the fact that realistic scanpaths do not follow ideal mathematical patterns. All of the figures provided in Figure 6 are classified using Procedure 4. Part (a) is classified by Step 1, part (b) is classified by Step 2, part (c) is classified by Step 3, and part (d) is classified by Step 4. The example in (c) is mixed because the first 10 fixations qualify as being circular since the θ values are increasing, and the last 10 fixations (excluding the third point) qualify as being linear since the X values are consistently decreasing as the Y values switch direction twice. The example in (d) is "other" because neither circular nor linear patterns occur consecutively for at least half of the aircraft.

3. Experiment

3.1. Participants. At Indianapolis ARTCC, 25 expert ATCs with FAA certification provided the scanpaths. Their experience ranged from 3 to 30 years with an average of 20.7 and standard deviation of 7.1. Due to too much loss of data to draw confident conclusions, 1 participant was excluded from each scenario resulting in the analysis of 24 participants across 3 scenarios for a total of 72 recordings.

3.2. Apparatus. A Tobii X60 eye tracker was used to collect the eye tracking data of the participants at a collection rate of 60 Hz. Simscope/Simtarget software was used to simulate a radar display of air traffic on a 48.26 cm LCD monitor. The eye tracker had an accuracy of 0.5° with each degree corresponding to approximately 1.2 cm when eyes were 68.6 cm from the monitor. Participants' eyes were about 100 cm from the monitor which resulted in a maximum fixation error of 1 cm. Data blocks were 1.5 cm by 1.1 cm; therefore the data block visual angle was about 0.6° due to the height. Digital surveillance radar (DSR) mode was used on Simscope/Simtarget with a refresh rate of 5 s, which simulates realistic radar displays with considerable accuracy.

3.3. Task. ATCs had to identify and solve conflicts while their eye tracking data was collected. Before the tasks began, 2 practice scenarios were performed in order to familiarize the ATCs with the simulation. During the following tasks that were recorded for data analysis, they were required to announce aircraft call signs of LOS pairs until no more remained during 3 unique scenarios.

3.4. Scenarios. There were 3 en route scenarios presented to each participant that are displayed in Figure 7: (1) low congestion scenario with 12 aircraft shown in (a), (2) moderate congestion scenario with 16 aircraft shown in (b), and (3) high congestion scenario with 20 aircraft shown in (c). The simulations have black backgrounds with bright green objects and text, but to enhance the images, the color was inverted and converted to black and white, and the text size was increased by 200%. Each small diamond shape symbolizes an aircraft and the line projecting out indicates the direction of travel. The display provides a top-down view of the aircraft as if they are being observed from above, looking down towards the Earth's surface. The three lines of text by each aircraft is the data tag which lists the flight number, altitude, and speed, respectively. The aircraft in these scenarios are en route; hence each has constant altitudes indicated by the "C" following the altitude.

3.5. Data Analysis. The independent variable was the aircraft congestion of low, moderate, and high. The dependent variable was each resulting scanpath. From each raw scanpath, the IGS was obtained and then extracted and fundamental IGSs were obtained. The raw scanpath was measured for total time, the IGS was analyzed for scan time and number of comparisons by applying Procedure 1, and the extracted and fundamental IGSs were analyzed for pattern classification of circular, linear, mixed, or other using Procedures 2–4 with interrater agreement.

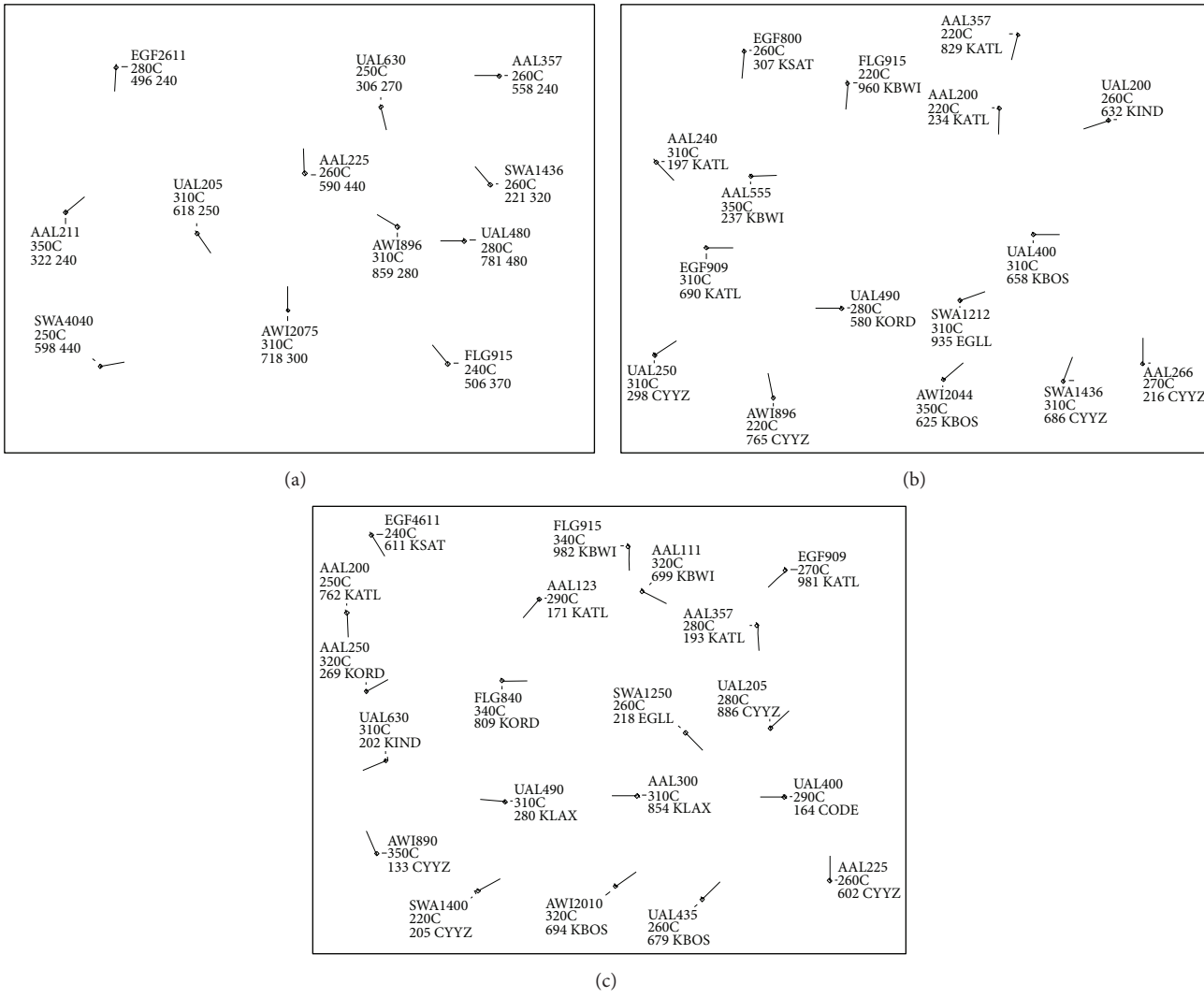


Figure 7: Scenarios performed by each participant. (a) Low congestion scenario. (b) Moderate congestion scenario. (c) High congestion scenario.

Tobii Studio software was used to collect and analyze the eye tracking data. The velocity threshold identification (I-VT) algorithm [34, 35] was used with the defaulted threshold of 0.42 pixels/ms to define a spatial fixation.

Analysis of oculomotor trends included average raw scanpath time versus aircraft congestion, average IGS time versus aircraft congestion, average number of IGS comparisons versus aircraft congestion, and average IGS time versus average number of IGS comparisons. The results were plotted and an ANOVA test was applied with pairwise comparisons between each relationship.

For visual scan pattern classification, two raters utilized Procedures 2–4 to reach an interrater agreement. Extracted scanpath classification is similar to the previous method used in [9] because interrater agreement was dominant and the procedures were only used as guidelines. Fundamental scanpaths were classified using a more elaborate procedure than what was previously used in [9]. Classification depended on the procedures; then interrater agreement was used to confirm accurate use of the procedures and reassign classification to exceptional scanpaths that were judged incorrectly classified by the procedures. Each scan was reviewed at least 3 times by the raters to minimize judgement errors. The results of extracted and fundamental scanpath patterns were compared against each other to see if they differed and with ATCs' verbal inputs from [21] to check consistency.

4. Results

From all 75 recordings, 3 were excluded due to missing periods of eye tracking data: participant 5 in low and moderate congestion scenarios and participant 22 in high congestion scenario. Therefore 72 recordings were used, 24 recordings from each scenario.

4.1. Oculomotor Trends. The oculomotor trends include scan time and number of comparisons as aircraft congestion increases. Figures 8(a) and 8(b) illustrate how scan time and

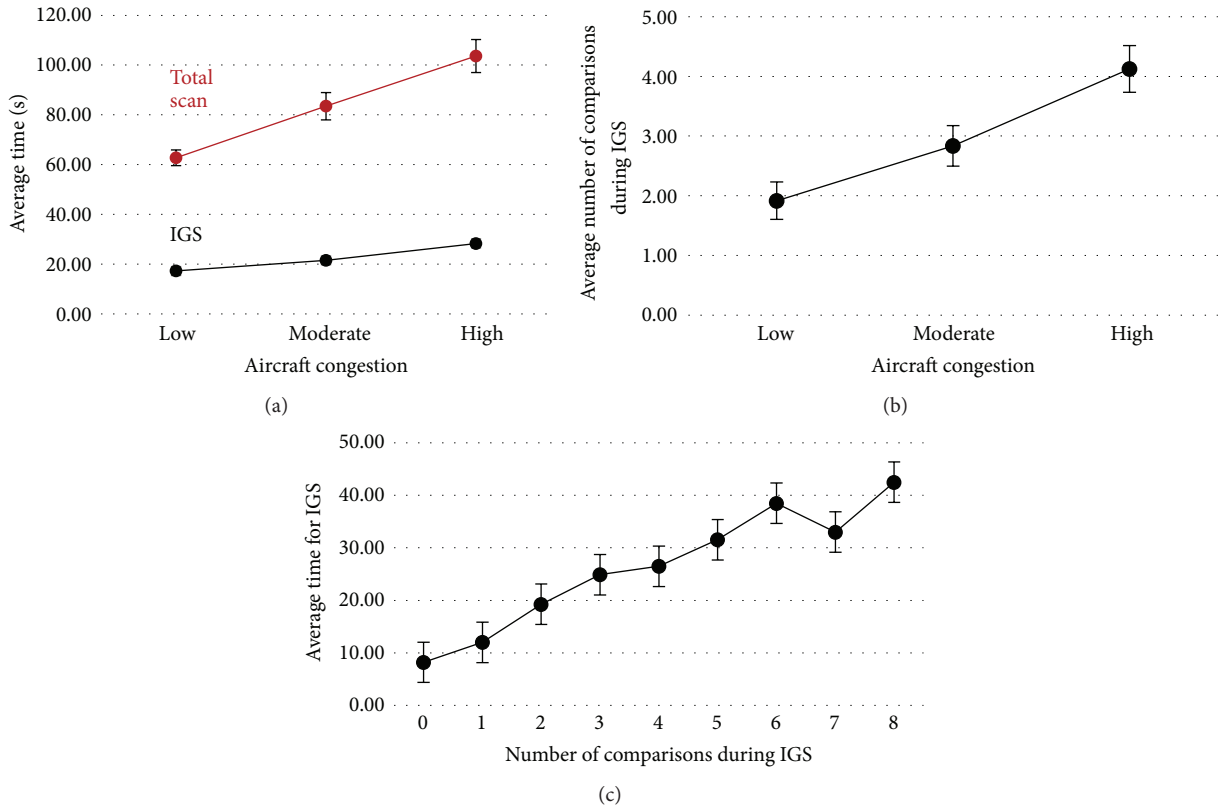


Figure 8: Oculomotor trends. (a) Average time to complete the raw scanpath (red) and average time to perform the IGS (black). (b) Number of comparisons during the IGS. (c) Average time to complete the IGS with respect to the identified number of comparisons during the IGS.

number of comparisons both increased with more aircraft. As ATCs conducted more comparisons, they took longer to complete the IGS as shown in Figure 8(c), which implies that increasing the amount of aircraft increases the number of comparisons which increases the required scan time.

For the following data analysis on oculomotor trends, $\alpha = 0.05$ was used to determine significance of results. The different congestion levels (low, moderate, and high) had significant effect on the total scan time ($F = 5.47$, $p < 0.001$) illustrated in Figure 8(a). Post hoc analysis (Tukey test) showed that there were significant differences among all congestion levels ($p < 0.001$ for all pairwise comparisons). Similarly, the different congestion levels had significant effect on the IGS time ($F = 4.14$, $p < 0.001$). Post hoc analysis (Tukey test) showed significant differences for low versus high ($p < 0.001$) and moderate versus high ($p < 0.001$), and marginal differences for low versus moderate ($p = 0.08$). The different congestion levels had significant effect on the mean number of comparisons ($F = 2.53$, $p < 0.001$) illustrated in Figure 8(b). Post hoc analysis (Tukey test) showed significant differences for low versus high ($p < 0.001$) and moderate versus high ($p = 0.010$), and insignificant differences were found for low versus moderate ($p = 0.108$). The number of comparisons had significant effect on the IGS time ($F = 11.80$, $p < 0.001$) illustrated in Figure 8(c). Post hoc analysis (Tukey test) showed significant differences for most pairwise comparisons, as depicted in Table 3.

4.2. Scanpath Patterns. The results of the extracted and fundamental scanpath classifications for different levels of congestion are provided in Figure 9. The number of participants to use the given patterns is shown for circular (C), linear (L), mixed (M), and other (O). Several trends can be witnessed from the data. The fundamental scanpaths show a similar trend for low and moderate congestion if mixed patterns are not considered (since it is unknown if they should be counted as circular, linear, or other): circular scanpaths are most common, followed by linear, and other scanpaths are least common. For high congestion, the pattern occurrences are fairly consistent. However, for the extracted scanpaths, the trends were quite different. Other scanpaths were most common due to the influence of local scans, and they in fact consist of almost half of the identified patterns. Circular patterns are slightly more occurrent than linear, but neither is frequent, and mixed patterns are least popular except in the moderate congestion scenario where they suddenly rise.

The detailed scanpath patterns identified from the extracted and the fundamental scanpaths of all participants during the IGS are shown in Table 4 for each scenario. As previously explained, the "extracted scanpath pattern" was determined by observing the IGS while excluding comparisons. The "fundamental scanpath pattern" was derived from the shape created by the order the aircraft were viewed, which excluded any repeated fixations. The fundamental scanpath is the IGS without local scans (which also excludes

Table 3: Least squares means for effect number of comparisons. Pr > |t| for $H0$: LSMean(i) = LSMean(j). Dependent variable: IGS time.

i/j	1	2	3	4	5	6	7	8	9
1		0.234	0.010	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
2	0.234		0.966	0.043	0.003	<0.001	<0.001	<0.001	<0.001
3	0.010	0.966		0.171	0.016	<0.001	<0.001	0.001	<0.001
4	<0.001	0.043	0.171		0.973	0.038	0.006	0.041	<0.001
5	<0.001	0.003	0.016	0.973		0.400	0.030	0.094	0.001
6	<0.001	<0.001	<0.001	0.038	0.400		0.360	0.906	0.012
7	<0.001	<0.001	<0.001	0.006	0.030	0.360		1.000	0.943
8	<0.001	<0.001	0.001	0.041	0.094	0.906	1.000		0.695
9	<0.001	<0.001	<0.001	<0.001	0.001	0.012	0.943	0.695	

Table 4: Scanpath patterns identified during IGS.

Participant	Extracted scanpath pattern			Fundamental scanpath pattern		
	Low	Moderate	High	Low	Moderate	High
1	Other	Mixed	Linear	Linear	Mixed	Linear
2	Other	Mixed	Mixed	Circular	Mixed	Mixed
3	Other	Circular	Mixed	Other	Circular	Circular
4	Circular	Circular	Circular	Circular	Circular	Circular
5	N/A	N/A	Linear	N/A	N/A	Linear
6	Other	Other	Other	Circular	Other	Other
7	Mixed	Linear	Linear	Mixed	Linear	Linear
8	Other	Mixed	Other	Mixed	Mixed	Mixed
9	Mixed	Linear	Linear	Mixed	Linear	Linear
10	Mixed	Other	Circular	Linear	Mixed	Circular
11	Other	Circular	Other	Linear	Circular	Linear
12	Circular	Other	Other	Circular	Circular	Other
13	Other	Other	Circular	Other	Circular	Circular
14	Circular	Other	Circular	Circular	Mixed	Circular
15	Other	Circular	Mixed	Mixed	Circular	Mixed
16	Other	Mixed	Mixed	Circular	Circular	Mixed
17	Linear	Mixed	Linear	Linear	Linear	Linear
18	Linear	Mixed	Other	Linear	Other	Other
19	Other	Mixed	Other	Circular	Mixed	Other
20	Linear	Linear	Other	Mixed	Linear	Other
21	Linear	Other	Other	Linear	Circular	Linear
22	Other	Other	N/A	Other	Other	N/A
23	Circular	Circular	Circular	Circular	Circular	Mixed
24	Circular	Other	Other	Circular	Other	Other
25	Mixed	Other	Circular	Mixed	Other	Circular

comparisons); consequently the scanpath does not return to aircraft already scanned. Usually the patterns observed in the fundamental scanpaths were the same or simpler than the extracted patterns, although that was not always the case; there were 3 exceptions (low scenario participant 20, moderate scenario participant 18, and high scenario participant 23). N/A indicates substantial loss of eye tracking data not included in the analysis.

Figure 10 illustrates the fundamental scanpaths. Each black dot represents an aircraft and the starting and ending aircraft are marked with a green star and red square, respectively. They are grouped into the categories exactly as indicated from Table 4 (C, L, M, or O) and the participant number is on the top left of each scanpath. The classification was chosen based on Procedures 2–4 provided above and then interrater agreement to reassign any exceptions. Note that the fundamental scanpaths are a still image drawn based on the initial location of each aircraft to provide better insights to the visual patterns. In reality, each aircraft is in movement during the IGS of ATCs. However, the movement of each aircraft was substantially small during the IGS (e.g., approximately 95 pixels or 0.25 cm of movement on the display every 5 seconds when flying at 300 knots). Therefore, it was determined that it was sufficient to use a single figure

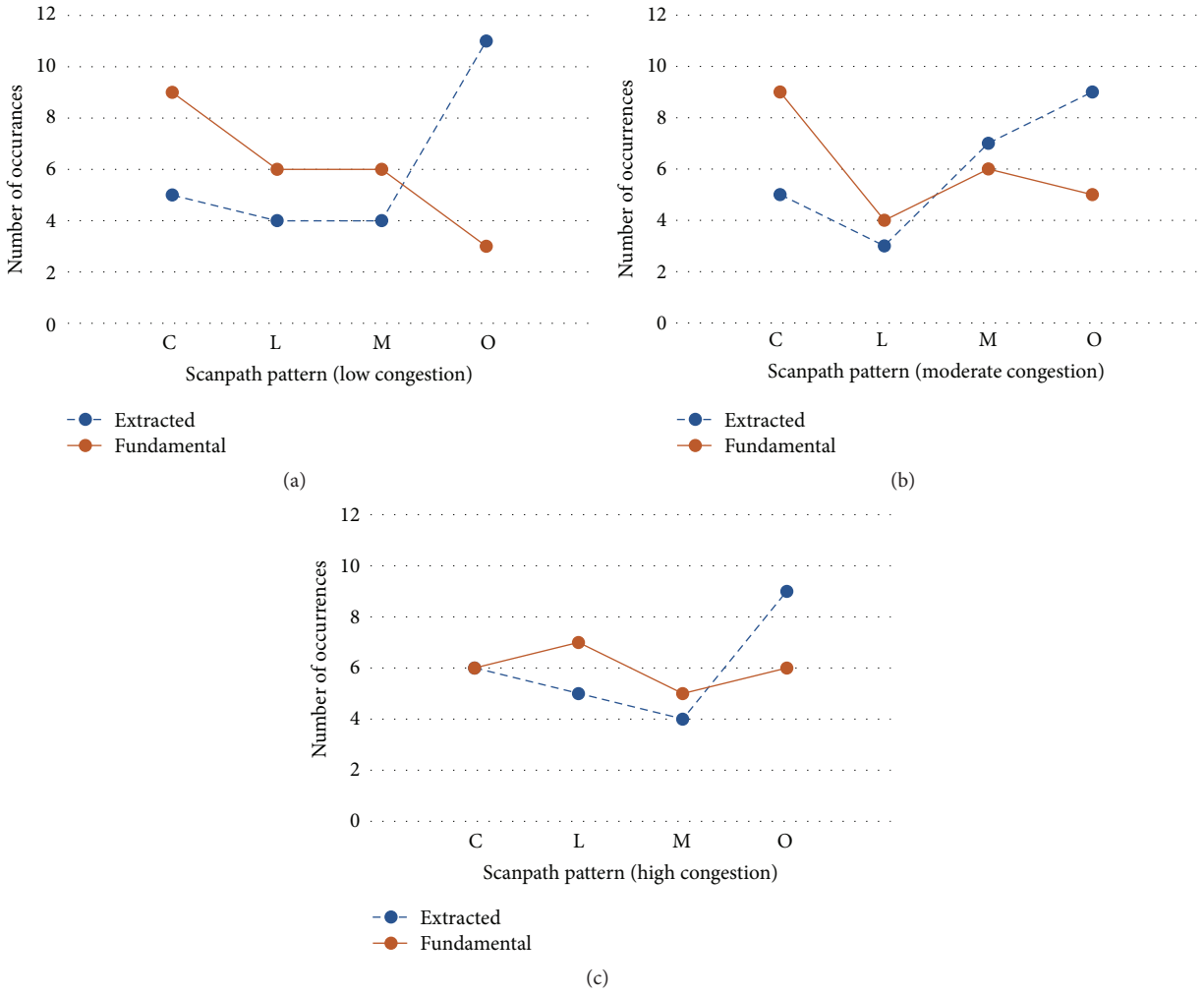


Figure 9: Number of participants classified with given scanpath patterns for extracted IGS (blue dashed line) and fundamental IGS (orange solid line). (a) Low congestion scenario. (b) Moderate congestion scenario. (c) High congestion scenario.

with initial aircraft location overlaid with the scanpath, rather than having to show multiple screenshots of different aircraft locations during the IGS.

Extracted scanpaths are not graphically displayed because they depended on the raters' cognitive ability of watching the IGS and disregarding any comparisons, similar to the previous method used for identifying scanpath patterns. Nonetheless, examples of identified global scanpaths from collected data are shown in Figure 11 when definite circular or linear patterns occurred. The aircraft and data tags are in green, the white numbered circles show the sequential eye fixations, and each line represents a saccade between fixations. Rater judgement was made on extracted scanpaths by observing the IGSs similar to global scanpaths displayed in Figure 11, cognitively neglecting comparisons, and then determining the classification based on the scanpath pattern.

The obtained results were compared to the ATCs' linguistic inputs from [21] in Table 5. The verbal inputs were applied to 26 low congestion scenario cases and only consisted of circular (C), linear (L), or other (O) patterns. The fundamental and extracted pattern results were applied to all scenarios and consisted of 24 cases each, with an additional mixed (M) category. As the table indicates, the low congestion fundamental patterns are most consistent with the ATCs' inputs. The occurrences from the results were expected to differ because of the added presence of a mixed category, but most of the trends are similar. If the mixed category was removed causing mixed scanpaths in the low scenario of fundamental patterns to instead be classified as 5 other patterns and 1 circular pattern, those findings would match the verbal inputs as much as possible given the unequal participant numbers.

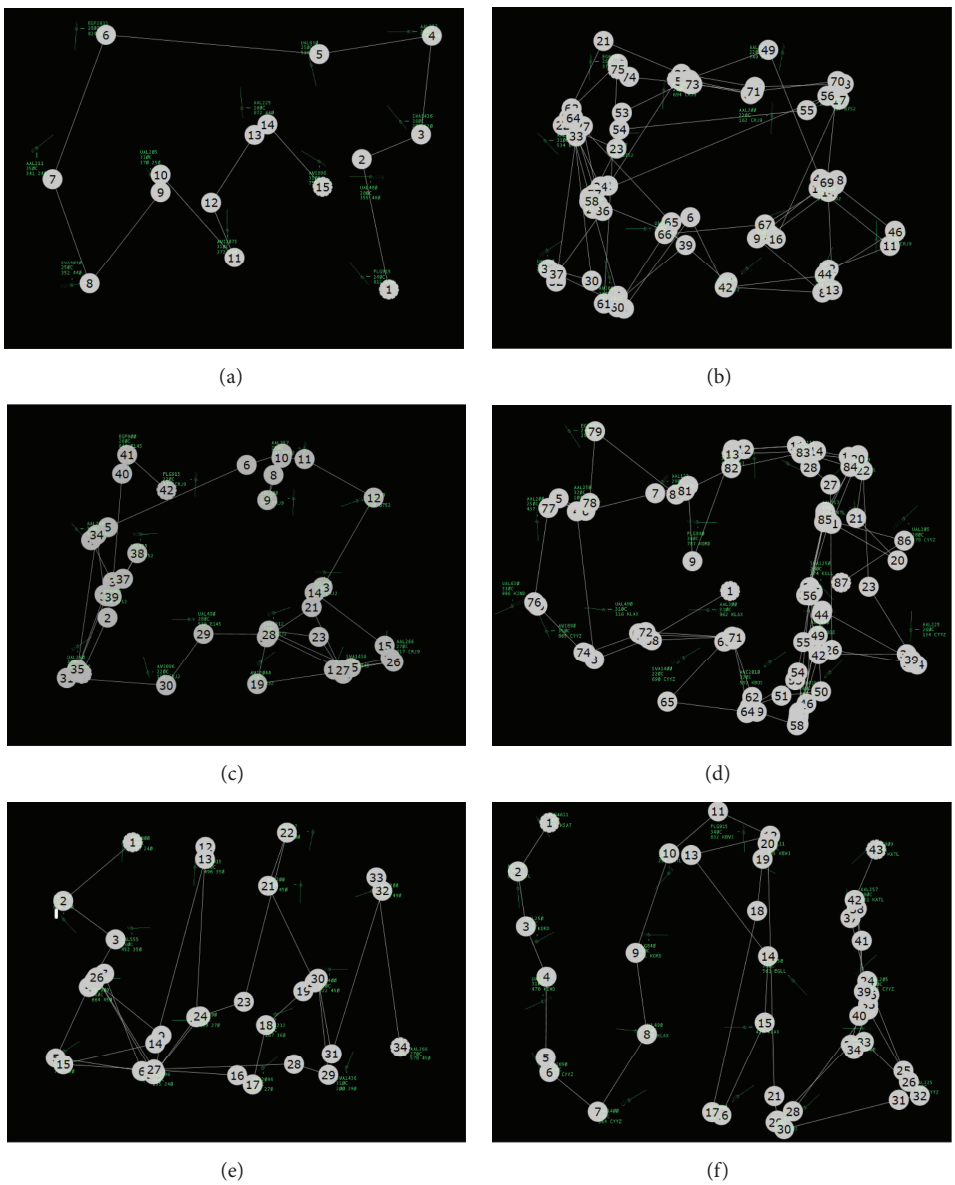
5. Discussion

The scanpaths were characterized based on (1) oculomotor trends including raw scanpath completion time, IGS time, and number of comparisons for differing aircraft congestion scenarios and (2) extracted and fundamental IGS patterns. Not all of the patterns that ATCs self-reported in [21]

Figure 10: Illustrated fundamental IGSs grouped in their pattern classifications. (a) Low congestion scenario. (b) Moderate congestion scenario. (c) High congestion scenario.

were identified, although similar patterns were found from observing the scans, without considering the linguistic input from the ATCs.

The oculomotor trends in Figure 8 showed that there were significant differences between congestion levels when examining total raw scanpath time or initial global scanpath (IGS) time. The scan times significantly increased as the congestion increased; however the amount of increase in the IGS time based on congestion was not proportionally linear to the amount of increase in total scan time. It appears that



(a) (b) (c) (d) (e) (f)

Figure 11: Examples of global scanpaths overlaid on the radar display. (a) Circular scanpath in low congestion scenario. (b) Circular scanpath in moderate congestion scenario with three complete revolutions. (c) Circular scanpath in moderate congestion scenario. (d) Circular scanpath in high congestion scenario. (e) Linear scanpath in moderate congestion scenario. (f) Linear scanpath in high congestion scenario.

the ATCs took much more time to detect the aircraft conflicts as the congestion level increased but tried to complete the IGS as quickly as possible. This reasoning is supported by the average rate of increase of the total raw scanpath time being higher than the average rate of increase of the IGS time. This is most likely due to the average number of comparisons during the IGS moving from only 2 (low congestion) to 3 (moderate congestion) to 4 (high congestion) comparisons; only 1 more comparison was used each time the congestion scenario increased, but the total number of comparisons probably increased much more across the scenarios which resulted in the total scan time rate increasing much faster than the IGS rate.

The visual scanning strategies show that the most dominant patterns used by expert ATCs were circular method followed by linear method, followed by other methods, which accords with the ATCs' linguistic inputs that were provided in [21]. An important finding was that the fundamental scanpaths showed more consistent matching with the ATCs' linguistic inputs compared to those identified from the extracted scanpaths. In Figure 9, similar trends can be viewed between circular, linear, and mixed patterns, but extracted scanpaths have much more "other" patterns than fundamental scanpaths. This indicates that the "other" patterns in extracted scanpaths were classified as being either circular, linear, or mixed in the fundamental scanpaths. When

Table 5: Pattern occurrence from ATCs' linguistic inputs compared to fundamental and extracted scanpaths.

	ATCs' inputs		Fundamental patterns						Extracted patterns					
	Low		Low		Moderate		High		Low		Moderate		High	
C	11	42%	9	38%	9	38%	6	25%	5	21%	5	21%	6	25%
L	6	23%	6	25%	4	17%	7	29%	4	17%	3	13%	5	21%
M	N/A	N/A	6	25%	6	25%	5	21%	4	17%	7	29%	4	17%
O	9	35%	3	13%	5	21%	6	25%	11	46%	9	38%	9	38%

observing the individual trends in Table 4, approximately 68% of the classifications were identically matched between those scanpaths, and the remaining 32% of pattern classifications consisted of the mentioned differences between the scanpaths.

Interestingly, there was a slight increase in the frequency of the linear search patterns when the congestion level was high, and possible reason for that could be due to the initial spatial layout of the multiple aircraft in that scenario which seemed to be dominantly linear. Based on the multiple observations of the ATCs' visual scanning patterns, the ATCs seem to apply an overall scan pattern (such as circular or linear), but it also seems that they move from one aircraft to another in close (or closest) proximity. If the spatial layout is somewhat linear, then even if an ATC has a circular search strategy in mind, the visual scanpath may result in a linear pattern and that could explain the higher number of linear patterns used in the high congestion scenario that can be seen Figure 9(c).

Another explanation could be that the ATCs may have changed their strategies from circular to linear as the congestion level increased. It may have been easier for the ATCs to use a linear scanning strategy to keep track of the observed aircraft as the scenarios became more complex. Individual scanpath pattern comparisons in Table 4 show that many ATCs were consistent with their visual scanning strategies across the scenarios (e.g., participants 4, 23, and 17); however, some ATCs showed different patterns among different congestion levels (e.g., participants 10, 11, and 25). However, the amount of change from circular to linear was not drastic, and it is challenging to identify the possible reasons of the individual inconsistencies by only examining the scanning patterns.

Figure 10 illustrates the categorized fundamental IGSs that were judged and agreed upon the interraters based on the developed processes. The result shows that scans can be classified by simply observing their filtered representations and applying definitive procedures without any follow-up validations by the ATCs. The obtained results could not be 100% mapped to the ATCs' linguistic inputs but showed high promise with similar mapping percentages, as shown in Table 5. Note that perfection was not expected since the results included an extra classification category (mixed) as opposed to those verbally expressed by ATCs, but the trends are similar in which circular patterns are more popular than linear patterns.

As previously mentioned, fundamental and extracted patterns differ mainly in the other category; other pattern is the most popular pattern for extracted scanpaths, but they are less frequent than circular patterns for fundamental scanpaths. The mapping percentages were closest to ATCs' inputs for the fundamental scanpath during the low congestion scenario. If the 6 occurrences that were judged mixed in that case were instead classified as 5 other and 1 circular, the trend would have matched the ATCs' inputs as much as possible (note that the ATCs' inputs total 26 while each fundamental and extracted scenario total 24). In fact, if all 6 of the mixed patterns were classified instead as other, the low congestion fundamental case would be consistent with the inputs provided in [21]. This consistency indicates promise in utilizing the classification method on fundamental scanpaths.

Limitations and Future Directions. Scanpath patterns used by ATCs can be unintended, incomplete, and limited to local scans, or overlapping with other categories which makes classification challenging. Circular and linear scans were the easiest to identify because they are independent of the above limitations mentioned; they depend on shape and follow certain procedures. However, regional, augmented, density-based, and proximity-based scans are considered difficult to identify since they do not depend on shape. At this time, they lack confident identification from the eye tracking data alone unlike circular and linear scans; therefore they hold too much uncertainty for current identification and are classified as other scans. Other and mixed scans are subjectively the most difficult to recognize, with other scans ranking as most difficult. Although the remaining patterns increase in classification difficulty, it remains useful to develop algorithms that can encapsulate all strategies mentioned by ATCs.

The effects of several variables have been studied in this work and in previous research, such as scenario congestion and difficulty. Spatial layout also needs to be investigated to determine how different layouts can influence the visual scanning patterns and analysis of them. In an extreme case, if all aircraft were aligned into a single line, the visual scanning pattern would always be linear even if the ATC attempted a circular strategy.

Perhaps the most difficult aspect of this research was in identifying the cognitive reasons to the observed visual scanning patterns under different aircraft congestion levels. The visual scanpaths provide different types of search methods but do not necessarily show the rationale underlying the search pattern. Therefore, a mixed method approach was required to validate the classified visual scanning patterns through the ATCs' follow-up confirmation on the classified patterns.

Based on this research, it seems that the ATCs' intended strategies are composed of 3 parts: (1) aircraft are searched with a pattern to complete a global scan, (2) aircraft in potential conflict are selected with local scans, and (3) comparisons are made between aircraft to solve conflicts [21]. The order in which these steps occur and whether they overlap differ with individual ATCs. For example, some ATCs completed a global scan before using local scans and then start comparisons, and others use local scans with comparisons to eventually form their global scan. If an ATC decides how to complete each step and in which order, it may be possible to define the ATC's intended strategy leading to better mapping the visual scanpaths to each ATC's intended strategy.

The goals in analyzing expert ATC scanpaths are to (1) develop high quality training programs for novices and (2) use automation to aid ATCs as their jobs grow more difficult with increased aircraft traffic. Characterizing ATCs' strategies observed during complex and critical situations can be used to better aid novice ATCs during training. Automation can be applied in many ways, such as informing ATCs when multiple aircraft were not scanned in an effective manner or when possible conflicting aircraft were not adequately identified. The succeeding step of this research is to compare the results obtained using the procedures on fundamental scanpaths with ATCs' inputs to test methodology accuracy through implementing the procedures into robust computer algorithms. Furthermore, in the long term, we should be able to support multimodal input analysis, such as corroborating EEG analysis with eye tracking analysis [36] to better support our goals.

6. Conclusion

Finding similar visual scanpath patterns that map with the ATCs' linguistic inputs were accomplished by selectively using the IGS, extracting the fundamental representation, and applying Procedures 2–4 for classification that allowed less reliability on rater judgement. The development and classification of the fundamental scanpaths showed promise in better mapping the visual scanning patterns to the ATC linguistic inputs; it was found that the mixed patterns should instead be most likely classified as "other" patterns. Moreover, oculomotor trends revealed the effects of different aircraft congestion; as congestion increased, scan time and number of comparisons increased as well. Scanpath patterns were also affected by increasing aircraft congestion by a higher occurrence of "other" patterns in the fundamental scanpaths, although more studies are required to determine the cause. Improving the classification procedures and developing algorithms would be highly useful for identifying scanpath patterns more accurately. Once appropriate algorithms are generated, pattern identification can be automated and utilized in further understanding of ATC cognitive processes, effective training methods, and improvements of the ATC interface.

Competing Interests

There is no conflict of interests to declare.

Acknowledgments

The authors would like to thank the Air Traffic Control Association (ATCA) and the ATCs at Indianapolis ARTCC for their gratuitous support on performing the experiment. The Office of the Vice President of Research (VPR) at the University of Oklahoma partially funded the research and funded the analysis phase.

References

[1] H. Zeier, P. Brauchli, and H. I. Joller-Jemelka, "Effects of work demands on immunoglobulin A and cortisol in air traffic controllers," *Biological Psychology*, vol. 42, no. 3, pp. 413–423, 1996.

[2] J. M. Finkelman and C. Kirschner, "An information-processing interpretation of air traffic control stress," *Human Factors*, vol. 22, no. 5, pp. 561–567, 1980.

[3] V. Hopkin, "The impact of automation on air traffic control systems," in *Automation and Systems Issues in Air Traffic Control*, pp. 3–19, Springer, Berlin, Germany, 1991.

[4] Boeing Commercial Airplanes Markey Analysis, *Current Market Outlook 2015–2034*, 2015, http://www.boeing.com/resources/boeingdotcom/commercial/about-our-market/assets/downloads/Boeing_Current_Market_Outlook_2015.pdf.

[5] C. Niessen and K. Eyferth, "A model of the air traffic controller's picture," *Safety Science*, vol. 37, no. 2-3, pp. 187–202, 2001.

[6] P.-V. Paubel, P. Averty, and E. Raufaste, "Effects of an automated conflict solver on the visual activity of air traffic controllers," *The International Journal of Aviation Psychology*, vol. 23, no. 2, pp. 181–196, 2013.

[7] U. Metzger and R. Parasuraman, "The role of the air traffic controller in future air traffic management: an empirical study of active control versus passive monitoring," *Human Factors*, vol. 43, no. 4, pp. 519–528, 2001.

[8] J. B. Brookings, G. F. Wilson, and C. R. Swain, "Psychophysiological responses to changes in workload during simulated air traffic control," *Biological Psychology*, vol. 42, no. 3, pp. 361–377, 1996.

[9] Z. Kang and S. J. Landry, "Using scanpaths as a learning method for a conflict detection task of multiple target tracking," *Human Factors*, vol. 56, no. 6, pp. 1150–1162, 2014.

[10] S. Sadasivan, J. S. Greenstein, A. K. Gramopadhye, and A. T. Duchowski, "Use of eye movements as feedforward training for a synthetic aircraft inspection task," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Technology, Safety, Community (CHI '05)*, pp. 141–149, ACM, Portland, Ore, USA, April 2005.

[11] J. H. Goldberg and J. C. Schryver, "Eye-gaze determination of user intent at the computer interface," in *Eye Movement Research—Mechanisms, Processes, and Applications*, vol. 6 of *Studies in Visual Information Processing*, pp. 491–502, Elsevier, Philadelphia, Pa, USA, 1995.

[12] Z. Kang and S. J. Landry, "An eye movement analysis algorithm for a multielement target tracking task: maximum transition-based agglomerative hierarchical clustering," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 13–24, 2015.

[13] D. Noton and L. Stark, "Eye movements and visual perception," *Scientific American*, vol. 224, no. 6, pp. 35–43, 1971.

[14] D. Noton and L. Stark, "Scanpaths in saccadic eye movements while viewing and recognizing patterns," *Vision Research*, vol. 11, no. 9, pp. 929–942, 1971.

[15] E. Stein, *Air Traffic Controller Scanning and Eye Movements in Search of Information—A Literature Review*, Defense Technical Information Center, Fort Belvoir, Va, USA, 1989.

[16] B. Willems, R. Allen, and E. Stein, *Air Traffic Control Specialist Visual Scanning II: Task Load, Visual Noise, and Intrusions into Controlled Airspace*, Defense Technical Information Center, Ft. Belvoir, Va, USA, 1999.

[17] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics*, vol. 24, no. 6, pp. 631–645, 1999.

[18] A. Neal and P. J. Kwantes, "An evidence accumulation model for conflict detection performance in a simulated air traffic control task," *Human Factors*, vol. 51, no. 2, pp. 164–180, 2009.

[19] E. M. Rantanen and A. Nunes, "Hierarchical conflict detection in air traffic control," *The International Journal of Aviation Psychology*, vol. 15, no. 4, pp. 339–362, 2005.

[20] K. Rayner, "Eye movements in reading and information processing: 20 years of research," *Psychological Bulletin*, vol. 124, no. 3, pp. 372–422, 1998.

[21] Z. Kang, E. J. Bass, and D. W. Lee, "Air traffic controllers' visual scanning, aircraft selection, and comparison strategies in support of conflict detection," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1, pp. 77–81, 2014.

[22] S. A. Brandt and L. W. Stark, "Spontaneous eye movements during visual imagery reflect the content of the visual scene," *Journal of Cognitive Neuroscience*, vol. 9, no. 1, pp. 27–38, 1997.

[23] A. T. Duchowski, J. Driver, S. Jolaoso, W. Tan, B. N. Ramey, and A. Robbins, "Scanpath comparison revisited," in *Proceedings of the ACM Symposium on Eye-Tracking Research and Applications (ETRA '10)*, pp. 219–226, ACM, Austin, Tex, USA, March 2010.

[24] C. M. Privitera and L. W. Stark, "Algorithms for defining visual regions-of-interest: comparison with eye fixations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 970–982, 2000.

[25] F. Cristino, S. Mathôt, J. Theeuwes, and I. D. Gilchrist, "ScanMatch: a novel method for comparing fixation sequences," *Behavior Research Methods*, vol. 42, no. 3, pp. 692–700, 2010.

[26] P. Hejmady and N. H. Narayanan, "Visual attention patterns during program debugging with an IDE," in *Proceedings of the 7th Eye Tracking Research and Applications Symposium (ETRA '12)*, pp. 197–200, Safety Harbor, Fla, USA, March 2012.

[27] J. H. Goldberg and J. I. Helfman, "Scanpath clustering and aggregation," in *Proceedings of the ACM Symposium on Eye-Tracking Research and Applications (ETRA '10)*, pp. 227–234, Santa Barbara, Calif, USA, March 2010.

[28] G. Underwood, P. Chapman, N. Brocklehurst, J. Underwood, and D. Crundall, "Visual attention while driving: sequences of eye fixations made by experienced and novice drivers," *Ergonomics*, vol. 46, no. 6, pp. 629–646, 2003.

[29] S. K. Mannan, K. H. Ruddock, and D. S. Wooding, "The relationship between the locations of spatial features and those of fixations made during visual examination of briefly presented images," *Spatial Vision*, vol. 10, no. 3, pp. 165–188, 1996.

[30] S. Mathôt, F. Cristino, I. D. Gilchrist, and J. Theeuwes, "A simple way to estimate similarity between pairs of eye movement sequences," *Journal of Eye Movement Research*, vol. 5, no. 1, article 4, 2012.

[31] R. Dewhurst, J. Jarodzka, K. Holmqvist, T. Foulsham, and M. Nystrom, "A new method for comparing scanpaths based on vectors and dimensions," *Journal of Vision*, vol. 11, no. 11, p. 502, 2011.

[32] R. Dewhurst, M. Nyström, H. Jarodzka, T. Foulsham, R. Johansson, and K. Holmqvist, "It depends on how you look at it: scanpath comparison in multiple dimensions with MultiMatch, a vector-based approach," *Behavior Research Methods*, vol. 44, no. 4, pp. 1079–1100, 2012.

[33] Z. Kang and S. J. Landry, "Capturing and analyzing visual groupings of multiple moving targets in an aircraft conflict detection task using eye movements," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 54, no. 23, pp. 1906–1910, 2010.

[34] P. Olsson, *Real-time and offline filters for eye tracking [M.S. thesis]*, Department of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2007.

[35] O. V. Komogortsev, D. V. Gobert, S. Jayarathna, D. H. Koh, and S. M. Gowda, "Standardization of automated analyses of oculomotor fixation and saccadic behaviors," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2635–2645, 2010.

[36] A. N. Belkacem, S. Saetia, K. Zintus-Art et al., "Real-time control of a video game using eye movements and two temporal EEG sensors," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 653639, 10 pages, 2015.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 9172506, 14 pages
http://dx.doi.org/10.1155/2016/9172506

Research Article

EyeTribe Tracker Data Accuracy Evaluation and Its Interconnection with Hypothesis Software for Cartographic Purposes

Stanislav Popelka,[1] Zdeněk Stachoň,[2] Čeněk Šašinka,[2] and Jitka Doležalová[1]

[1] *Palacký University, Olomouc, 77146 Olomouc, Czech Republic*
[2] *Masaryk University, 61137 Brno, Czech Republic*

Correspondence should be addressed to Stanislav Popelka; standa.popelka@gmail.com

Received 25 November 2015; Revised 8 February 2016; Accepted 22 February 2016

Academic Editor: Ying Wei

The mixed research design is a progressive methodological discourse that combines the advantages of quantitative and qualitative methods. Its possibilities of application are, however, dependent on the efficiency with which the particular research techniques are used and combined. The aim of the paper is to introduce the possible combination of Hypothesis with EyeTribe tracker. The Hypothesis is intended for quantitative data acquisition and the EyeTribe is intended for qualitative (eye-tracking) data recording. In the first part of the paper, Hypothesis software is described. The Hypothesis platform provides an environment for web-based computerized experiment design and mass data collection. Then, evaluation of the accuracy of data recorded by EyeTribe tracker was performed with the use of concurrent recording together with the SMI RED 250 eye-tracker. Both qualitative and quantitative results showed that data accuracy is sufficient for cartographic research. In the third part of the paper, a system for connecting EyeTribe tracker and Hypothesis software is presented. The interconnection was performed with the help of developed web application HypOgama. The created system uses open-source software OGAMA for recording the eye-movements of participants together with quantitative data from Hypothesis. The final part of the paper describes the integrated research system combining Hypothesis and EyeTribe.

1. Introduction

The paper presents methodological-technical approach combining quantitative and qualitative methods which are based on specific technical tools. The aim of this paper is to introduce the newly developed technical research system and results of its validation: specifically, the creation and empirical verification of an interconnection of a web-based platform Hypothesis with an EyeTribe eye-tracking system connected to open-source software OGAMA. The interconnection was done by the creation of a new web application HypOgama.

The introduction of the paper discusses the methodology and mixed-research design (combination of quantitative and qualitative, resp., explorative methods) in the area of cognitive visualization and cartography. The paper consists of three parts which are ordered due the logic and procedure of the research system creation and verification. The first part is focused on the presentation of a tool for mass data collection: web-based platform Hypothesis. The second part of the paper presents the new low-cost eye-tracking system EyeTribe, which allows efficient realization of qualitative, respectively, explorative studies. In this part, close attention is paid to empirical study verifying the truthfulness of the low-cost EyeTribe tracker in comparison with SMI RED 250 system. The final part of the paper describes the research system which combines and integrates above-mentioned tools. Part of this last section is also an illustration of possible empirical study, where the interconnection of Hypothesis and EyeTribe for cartographic and psychology research is presented. However this case study is only an example of how the integrated

research system and HypOgama application works, and it should only illustrate the procedure of conducting a mixed-research design.

A significant portion of experimental studies in the area of cognitive visualization can be sorted into two main categories. The studies in the first category monitor and record the behaviour of individuals or, rather, their conscious actions and general work methods when completing tasks with a use of a map. The most common aspects of studies are completion speed, accuracy, and correctness or frequency of a given solution (see [1–5]). The mentioned studies use a quantitative approach and subsequent statistical methods of data analysis. A second significant category is the use of eye-tracking systems. Eye-tracking studies are in many cases combined with the recording of conscious behaviour, that is, user actions (see the first category), but the crucial activities recorded are eye-movements, which offer continuous data about (even unconscious) behaviour of the participant while solving a task. In other words, the focus of the user's attention is foregrounded [6]. Due to the high processing requirements, these studies are often performed on a small sample of participants and methods other than statistical data analysis are being used, for example, explorative data analysis [7].

Eye-tracking was used for the evaluation of maps for the first time already in the late 1950s [8], but it has been increasingly used in the last ten to fifteen years. The main reasons are the declining prices of the equipment and the development of computer technology that allows faster and more efficient analysis of measured data. For usability research, eye-tracking data should be combined with additional qualitative data, since eye-movements cannot always be clearly interpreted without the participant providing context to the data [9].

An example of comprehensive research in the field of cognitive visualization by using eye-tracking is the work of Alaçam and Dalcı [10], who compared four map portals (Google Maps, Yahoo Maps, Live Search Maps, and MapQuest). The basic assumption of the study was that lower average fixation duration indicates more intuitive map portal environment. The shortest average fixation duration was found in the case of Google Maps. Fabrikant et al. [11] used eye-tracking for the evaluation of map series expressing the evolution of the phenomenon over time, or for evaluation of user cognition of weather maps [12]. Ooms et al. [13] dealt with the suitability of map label positions and differences in map reading between experts and novices. Popelka and Brychtova [14] investigated the role of 2D and 3D terrain visualization in maps.

Olson [15] compared cognitive visualization and cognitive psychology, arguing that cartographers can adapt ideas and experiments in methodology from cognitive psychologists. Equally, psychologists can use maps as stimuli in their studies. Both disciplines can examine the cognitive processes while reading and understanding maps. However, cognitive psychologists are interested in different types of cognitive processes such as attention, visual perception, memorizing, or decision-making. A map is only a tool in this context. For a cognitive cartographer, the map is far more important.

The approach mentioned above is based on close cooperation between cartographers and psychologists and shows the possibility of a connection between large-scale studies and small-scale studies based on gathering and analysing eye-tracking data. Differences between large-scale and small-scale studies are described in Figure 1.

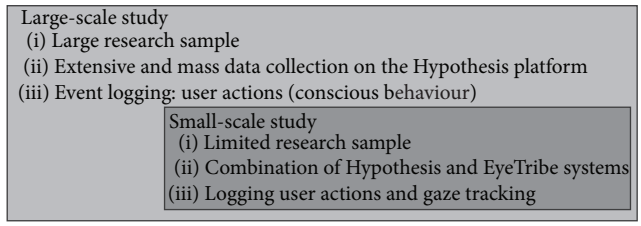


Figure 1: The combination of large-scale and small-scale study.

As it is discussed in Štěrba et al. [16], using only a qualitative (explorative) or quantitative type of evaluation method is not sufficient. Therefore, it is necessary to combine those methods, enabling their suitable completion, obtaining more valid results, and achieving better interpretation. A combination of quantitative and qualitative methods was established as mixed-research design [17]. The key idea and innovation of our method are the interconnection of two approaches in the area of cognitive visualization and also finding a technological solution.

The Hypothesis platform serves primarily for the creation of experimental test batteries, online administration, and extensive data gathering. After connecting with the eye-tracking system, more detailed data on the experimental task processing methods are gathered, which allow deeper insight into the postulated cognitive processes that underlie the behavioural reactions.

Štěrba et al. [18] propose two variants of mixed-research design:

(i) Using the eye-tracking system for a pilot study examining a quality of experiment design with results from this pilot study being used for improvement of experiment design before large-scale data collection.

(ii) Using Hypothesis for large-scale quantitative approach and secondary using of eye-tracking method for the subsequent specification of certain results with adjusted or changed types of tasks.

Both approaches and technical specification of Hypothesis platform are described in detail in [18] and are available online in English.

2. A Tool for Mass Data Collection: Web-Based Platform Hypothesis

For the purposes of large-scale experimental investigation, the creation of psychological tests, and evaluation of cartographic works, new research software concept was designed within the project "Dynamic Geovisualization in Crisis Management" [19]. Subsequently, this concept has been realized, and original software MuTeP was developed [20, 21]. MuTeP

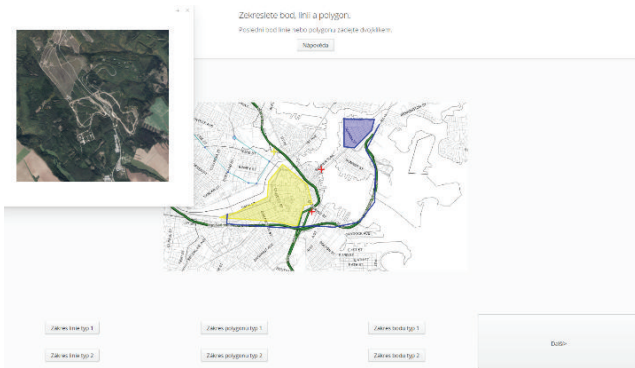


Figure 2: Example task on WMS interactive map. The user indicates the requested objects, draws lines, and marks out target areas by polygons. In the example shown, the user called up an orthophoto map in a dialogue-window. All the actions including the drawn point coordinates, lines, and polygons are saved in the database, and the correctness of the solution is automatically evaluated under preset conditions.

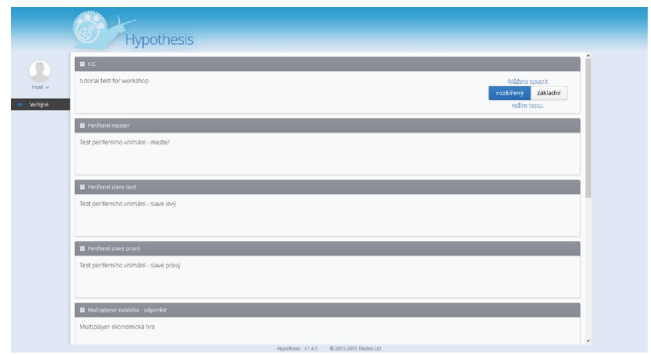


Figure 3: Management module in the Hypothesis platform. The user can launch the available tests in two modes: (a) legacy (launches in a normal browser) and (2) featured (launches in a controlled mode in SWT browser). The manager and the superuser have an extended access and can unlock the tests, create users, export results, and so forth.

was primarily created for the purposes of objective experimental exploration and evaluation of cartographic products in the perspective of user personality.

Although MuTeP was practically proven [22], it was clear that the conception used will soon reach its limits. Another impulse for the search for a more flexible solution was an effort to involve dynamic cartographic visualization as stimuli, randomization, nonlinear test batteries, connection with eye-tracking technology, and so forth, which were not possible to implement into MuTeP software.

Based on experience with MuTeP and in the context of current requirements, a new software concept was designed. This new software should have the potential for long-term growth and development [23]. Hypothesis has several important advantages in comparison with MuTeP. Above all, Hypothesis enables computer adaptive testing and offers a modular solution with plugin support (such as video or interactive animation plugins) and enables the work with interactive maps (such as web map services; see Figure 2).

The technology used for designing Hypothesis consists of the following: (1) the application core and user interface are built on framework Vaadin 7; work with the database is provided by ORM Hibernate; and (2) PostgreSQL in version 9.1 (and higher) is used as a primary database system [18].

The architecture of the system is three-layer: a client, server, and database. The client part is designed for communication and interaction with the user, and its operation is provided by standard web browsers (thin client) or a special browser distributed in the application package—special Hypothesis Browser. Hypothesis Browser is based on Standard Widget Toolkit (SWT) components and ensures more strict conditions and control over running tests [18, 24].

Hypothesis works as an event-logger application, which logs all user actions and events (coordinates and timestamps of clicks, key presses, start and end time of each presented slide, exposition time of every component such as a picture or dialogue-window, zoom of maps, rotation of 3D objects, etc.). Extensive logging of user actions and events is enabled through the structure of the final slides used for the test battery (package). The package comprises the hierarchical structure of branches which contain one or more tasks, and each task contains at least one slide. The slide consists of a template and content. Such structure enables nonlinear branching of the test slides or randomization of slides. All parts of the package are stored in structured XML format. After starting a test, a selected package is loaded from the database to the server application and a new test is created. Emphasis was placed on variability and range of software usability. Figure 2 shows an example of the slide using WMS. The slide consists of two layers. The underlaying image is created with a layer: ImageLayer. Above it, there is a transparent layer: FeatureLayer, which is designed to draw demanded points, polylines, or areas by mouse and store the events [18].

Hypothesis is also improved with two new key functionalities that are vital for the interconnection between eye-tracking systems (or other peripherals such as EEG) and enable the realization of experiments with high reliability. These functionalities involve the use of SWT browser that allows the client to monitor and control the testing process. In other words, when using the controlled mode (see Figure 3), the participant has no way to intentionally or unintentionally exit the test by, for example, pressing alt + F4. Other common functions of web browsers are also strictly disabled, such as page refreshing or opening menus by right-clicking the mouse. The second key functionality is the recording of two time sets in the database. To avoid the problem of slow internet connection, both server time and local PC time are recorded, which means that events on the client side can be accurately synchronized (e.g., synchronizing stimulus exposition with data from the eye-tracker).

Researchers can effectively create new test batteries thanks to a combination of a number of subfunctions and tools. Emphasis is also placed on the efficiency of the software. Researchers can effectively change the content of already finished test slides and create derivatives from sample

Table 1: Summary of calibration results for all participants.

Participant	SMI X	SMI Y	EyeTribe
P01	0,4	0,2	Good
P02	0,3	0,1	Poor
P03	0,4	0,6	Moderate
P04	0,4	0,4	Perfect
P05	0,9	0,5	Good
P06	0,3	0,5	Redo
P07	0,2	0,4	Moderate
P08	0,6	0,3	Moderate
P09	0,4	0,1	Perfect
P10	0,3	0,4	Poor
P11	0,6	0,3	Poor
P12	0,5	0,5	Moderate
P13	0,3	0,3	Moderate
P14	0,4	0,6	Poor

templates through the modules for user access administration and also export structured results.

Hypothesis software is freely available for collaboration on a various research topic in the Czech Republic and abroad. Access to the database and modules is provided after registration.

3. In-Depth Analysis of Cognitive Processes Using Eye-Tracking System

3.1. EyeTribe Tracker. Eye-tracking technology is becoming increasingly cheaper, both on the hardware and on the software front. Currently, the EyeTribe tracker is the most inexpensive commercial eye-tracker in the world, at a price of $99. More information about the device is available at the web page of the manufacturer (https://theeyetribe.com/). The low-cost makes it a potentially interesting resource for research, but no objective testing of its quality has been performed as of yet [25]. Dalmaijer in his study [25] with five participants compared the EyeTribe tracker with high-frequency EyeLink 1000. He states that concurrent tracking by both devices of the same eye-movements proved to be impossible, due to the mutually exclusive way in which both devices work. One of the reasons was that EyeLink uses only one eye for the recording. Delmaijer [25] also states that recording with both devices at the same time results in deterioration of results of both and often leads to a failure to calibrate at least one. Ooms et al. [26] compared EyeTribe with SMI RED 250 but also did not use the concurrent recording. In our study, we compared the EyeTribe tracker with SMI RED 250. In our case, we have not noticed any problems with calibration (see Table 1).

3.2. Methods of EyeTribe Accuracy Evaluation. For the comparison study, recording with SMI RED 250 and the EyeTribe tracker at the same time was performed. Laboratory setup is displayed in Figure 4. The EyeTribe tracker stands in front of the SMI device.

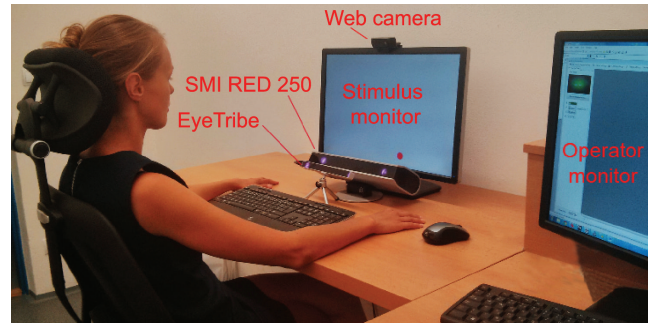


Figure 4: Laboratory setting for EyeTribe and SMI accuracy comparison.

EyeTribe tracker was connected with the OGAMA software [27], where the experiment with six static image stimuli was prepared. At the same time, screen recording experiment was created in SMI experiment center (sampling frequency was set up to 60 Hz, to be the same as EyeTribe). Both devices were calibrated separately (but the eye-trackers were at their positions and turned on).

After calibrations, recording with SMI started. After that, experiment with static images in OGAMA was performed. That means the SMI device recorded the experiment data as well (as a screen recording video). The whole experiment procedure was done with fourteen participants. The purpose of the study was to verify how trustworthy data from EyeTribe tracker are. Recorded fixations from both eye-trackers were compared qualitatively and quantitatively. A diagram of the whole recording procedure is displayed in Figure 5.

For the comparison of recorded data from both devices, the OGAMA environment was used. Data from EyeTribe were displayed in OGAMA directly; SMI data had to be converted. For this conversion, the tool smi2ogama developed by S. Popelka was used. The tool is available at http://eyetracking.upol.cz/smi2ogama/.

The recorded screen data were cropped according to the pertinence to individual stimuli. For that, recorded key presses (for a slide change) were used.

3.3. Participants. Total of 14 respondents participated in this part of the study (ten males and four females with an average age of 29.5). They were employees and postgraduate students of department of geoinformatics. 16-point calibration was used for both devices. Results of calibration are summarized in Table 1. With the EyeTribe, it was almost not possible to achieve perfect calibration result. Figure 6 shows the details of calibration results for participant P03. The results in OGAMA show calibration result for each of the 16 calibration points (with the use of colour); SMI shows only the average value in degrees of visual angle for axes X and Y.

For all recordings, I-DT fixation detection in OGAMA was used with the same settings. A value of 20 px was used as "maximum distance"; "minimum number of samples" was set up to 5. More information about fixation detection settings is available in [28, 29].

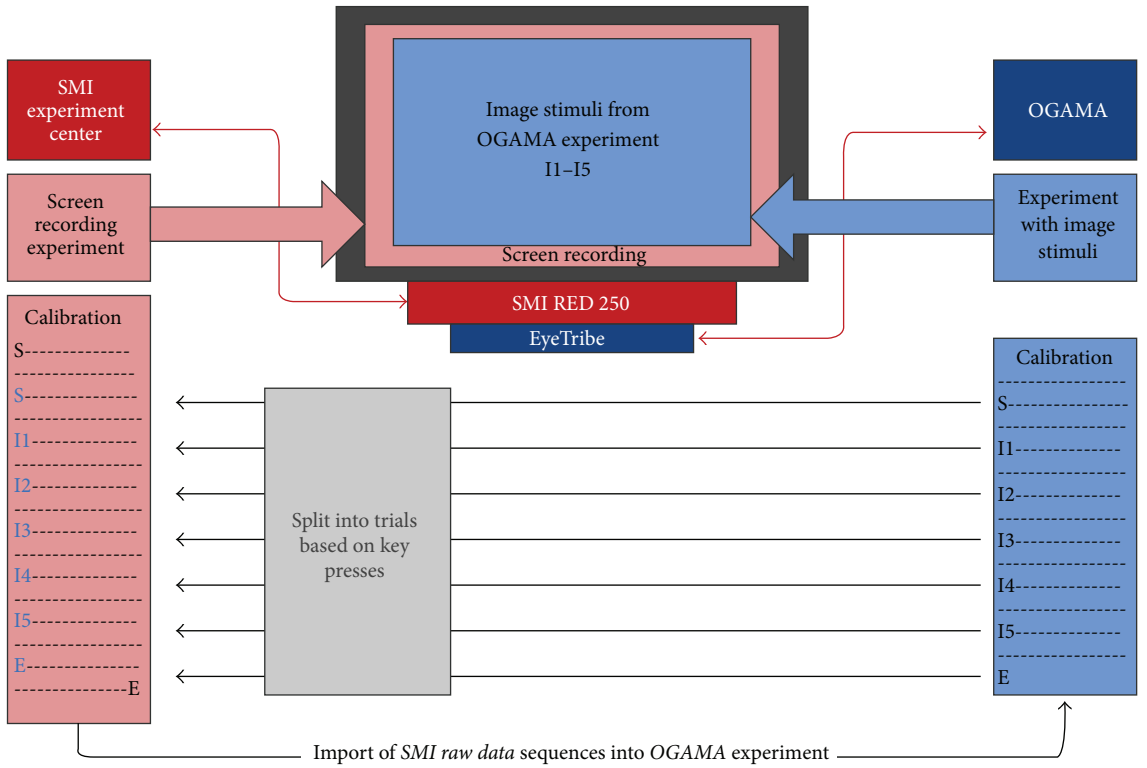
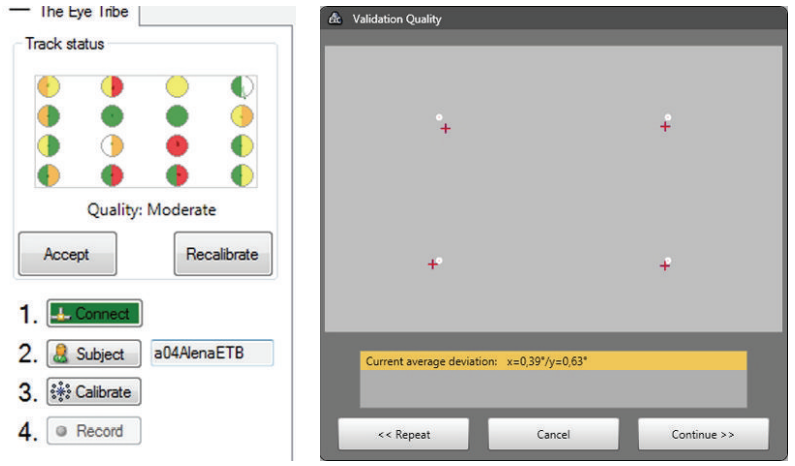


Figure 5: Diagram of concurrent eye-movements recording with SMI RED 250 and EyeTribe.



(a) (b)

Figure 6: Calibration results from EyeTribe (a) and SMI RED (b) for participant "P03."

3.4. Stimuli. The experiment contained six static images. The first one contained a grid with nine numbers; second one (Slide 2, Figure 7) contained sixteen numbers. The task of the participants was to read numbers in ascending order (from top to the bottom). Next three stimuli contained different types of maps, but the results of these stimuli are not described in this paper. The last stimulus (Slide 6, Figures 8 and 9) contained a map of the world and respondents' task was to move the eyes around Africa.

3.5. Results and Discussion of EyeTribe Evaluation. Eye-movement data recorded from participant P03 are displayed in Figure 7. Red points represent fixations from SMI, and blue points are fixations from EyeTribe. The task in this stimulus was only to read the numbers.

From Figure 7, it can be seen that both devices recorded around one or two fixations over each number. The accuracy of the recording is comparable. Accuracy reflects the eye-tracker's ability to measure the point of regard and is defined

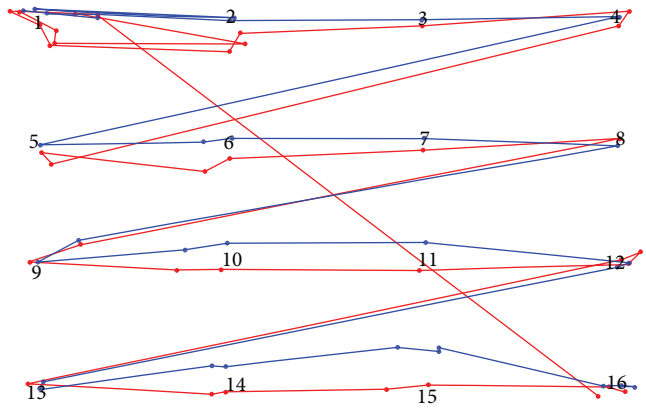


Figure 7: Comparison of recorded eye-movement data from participant P03 in Slide 2 from EyeTribe (blue) and SMI RED (red).

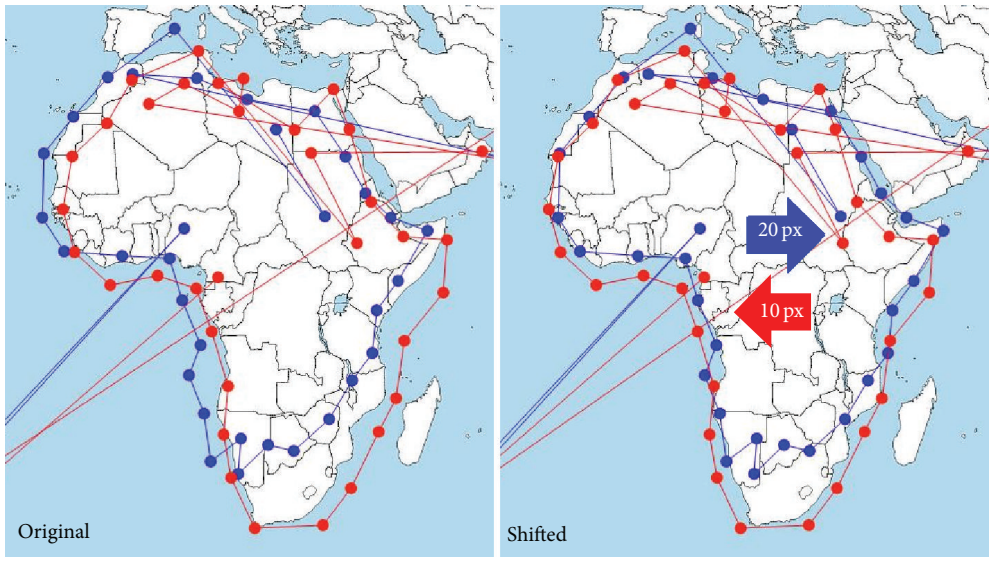


Figure 8: Comparison of recorded eye-movement data from participant P03 in Slide 6 from EyeTribe (blue) and SMI RED (red).

as the average difference between a test stimulus position and the measured gaze position [30]. The largest deviations of the EyeTribe tracker data were observed for two points in the middle of the bottom line. This situation was observed in almost all recorded data. The situation can be seen in Figure 7 in the case of points 14 and 15 (middle points in the lowest line of numbers). Gaze position recorded by EyeTribe is shifted upwards.

Another example is visible in Figure 8, which is the crop of Slide 6 stimuli. In this stimulus, the task was to move the eyes around the continent of Africa on the map. The data recorded by EyeTribe tracker were moved to the left by 20 px, but this systematic error can be corrected by a manual shift of fixations in OGAMA. This situation is depicted in Figure 8. On the left side, original data are displayed. On the right, data after horizontal shift (20 px to the right for EyeTribe and 10 px to the left for SMI) are depicted. Eye-movement data from EyeTribe for horizontally central fixations are shifted upwards, especially in the bottom part of the stimuli. See Figure 12 for more detailed analysis of fixation locations. The same issue was reported in all stimuli for most of the participants. Visualization of gaze trajectories of all participants is in Figure 9. The solution for dealing with this inaccuracy is to avoid placing important parts of the stimulus to the bottom of the screen. It will be possible to compare recorded raw data, but, in cartographic research, fixations are used for analysis, so it was more meaningful to compare fixations (identified with the same algorithm).

As an alternative for the comparison of raw data, comparison of data loss was performed. In the case of SMI recordings, average data loss (samples with coordinates 0, 0) was 0.57% of all recorded data. With the EyeTribe, the average data loss was 1.22%. Although the value is more than twice higher than in the case of SMI, it is still acceptable.

The graph in Figure 10 shows the percentage of data loss for Slide 2. It is evident that data loss is higher in the case of EyeTribe recordings, but, in most cases, less than 2% of data is missing. The highest values were observed for participants

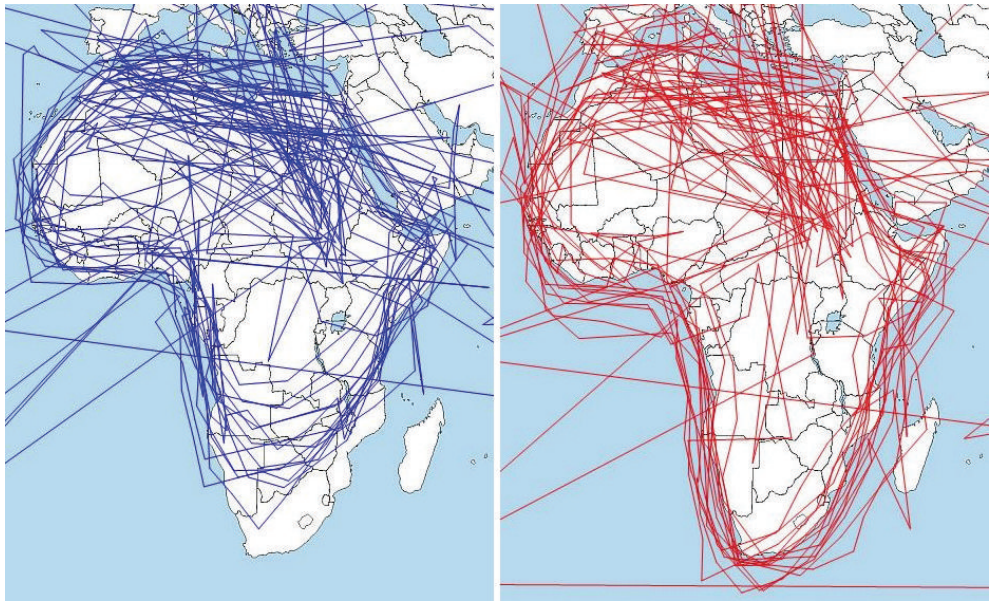


Figure 9: Problems with data recorded by EyeTribe (blue) at the bottom of the stimuli in comparison with SMI data (red).

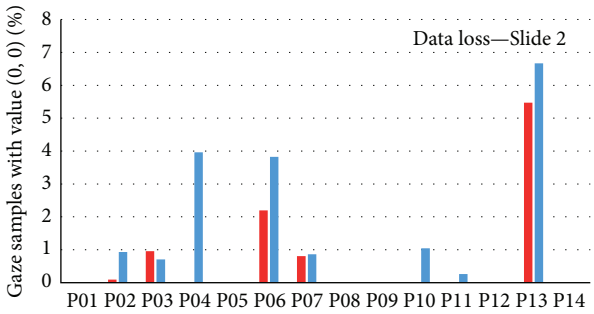


Figure 10: Comparison of data losses of fourteen participants during observation of Slide 2. Red bars represent SMI RED 250; blue ones represent EyeTribe tracker.

P06 and P13. Participant P06 had the worst calibration from all respondents. Participant P13 has worn glasses which can possibly cause the high data loss.

In the next step of accuracy evaluation, values of eye-tracking metric fixation count recorded by SMI RED 250 and the EyeTribe tracker were compared for all six stimuli in the experiment. A summary of the results is shown in Figure 11. The correlation between numbers of detected fixations was between 0.949 and 0.989 with the exception of participant P13 with the correlation of 0.808. The ratio between a number of recorded fixations with SMI device and EyeTribe was also investigated. On average, EyeTribe recorded 88.2% of fixations that were recorded by SMI device. The correlation and ratio values for each participant are presented as part of Figure 11.

Beside the number of fixations, their location was compared. For this evaluation, Slide 2 with a grid of 16 numbers was chosen (Figure 7). For each participant, the deviations between coordinates of the target (number) and closest fixation were calculated. The graphs in Figure 12 show the median size and direction of the deviation for each of the 16 targets in the stimuli. It is evident that the largest deviations (heading upwards) for EyeTribe were observed for the points in the bottom part of the image (numbers 14 and 15). Each graph contains the value of the Euclidean distance of median deviations from the origin. Average deviation was 26 px for EyeTribe and 22 px for SMI.

The evaluation of truthfulness was performed on fourteen participants. According to Nielsen [31], this number should be sufficient. The evaluation of qualitative (Figures 7, 8, and 9) and quantitative (Figures 10, 11, and 12) data indicates that accuracy of low-cost EyeTribe tracker is sufficient for the use in cartographic research. Similar results were found by Ooms et al. [26], who measured the accuracy by the distance between recorded fixation locations and the actual location.

The limitation of the low-cost device is the sampling frequency, which is only 60 Hz (compare with 250 Hz of SMI RED eye-tracker). Another problem is shift of fixation locations in the bottom part of the screen. Taking into account described limits of the device, the EyeTribe may be an appropriate tool for cartographic research.

4. Integrated Research System: Interconnection of Hypothesis Software and EyeTribe

As one of the practical applications of the mixed-research experiment design, the Hypothesis software interconnected with the EyeTribe tracker was chosen. For the recording of eye-tracking data, the OGAMA software was used because the EyeTribe tracker is intended for developers and contains no software for data recording and analysis. OGAMA has an inbuilt slide show viewer, but the range of functionality of this viewer in comparison with SW Hypothesis is quite

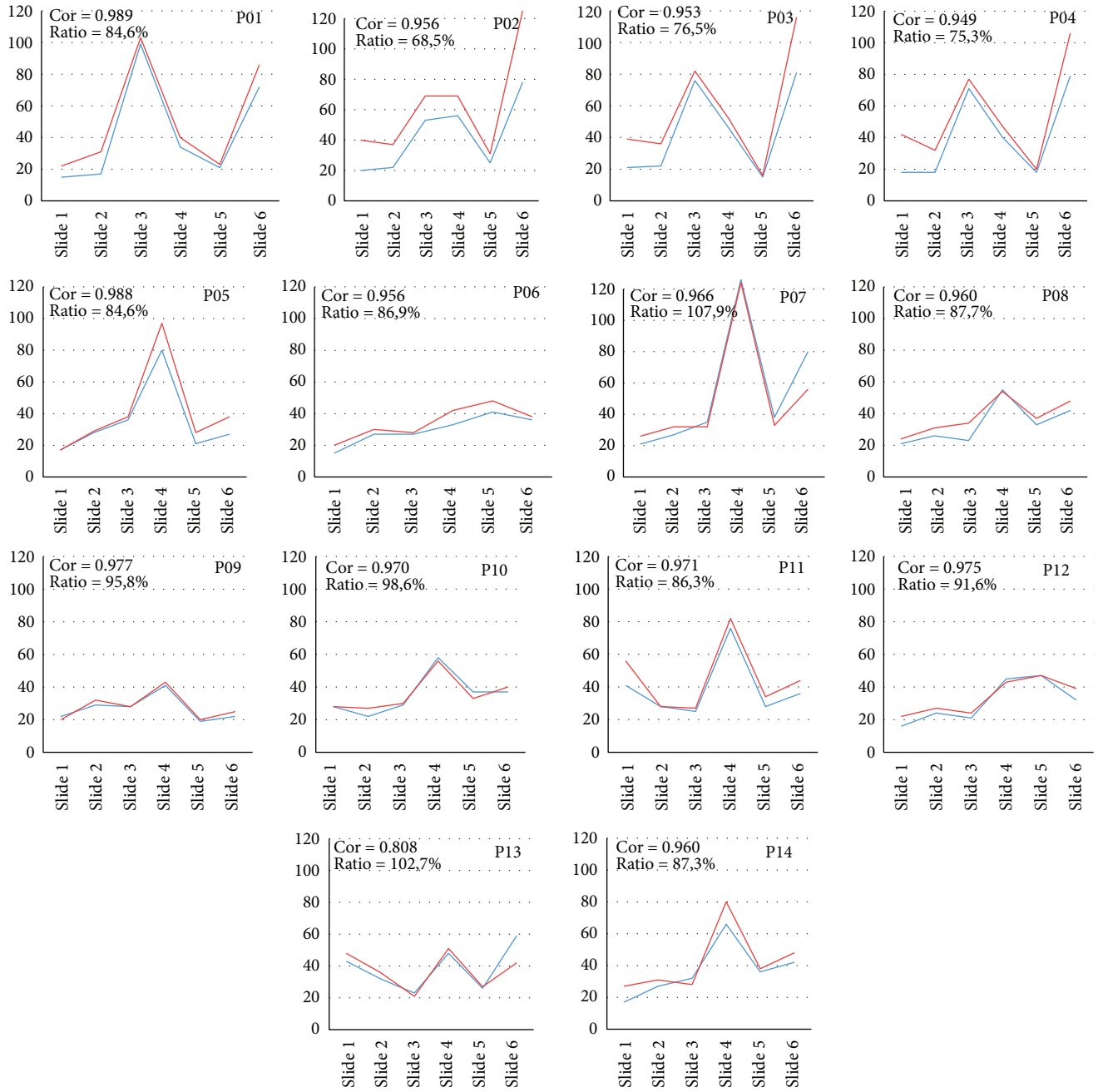


Figure 11: Comparison of fixation count eye-tracking metric for fourteen participants. EyeTribe data are displayed as blue line; SMI data are displayed as red line.

limited. Desktop application OGAMA principally does not allow working with web-based interactive maps and mouse clicks are recorded but not shown. Oppositely, Hypothesis visualizes clicks and allows drawing of lines and polygons. This functionality is crucial in the context of working with maps. Because of this functionality, Hypothesis connected to OGAMA via HypOgama was used.

4.1. Methods of Hypothesis and EyeTribe Interconnection. For the study, a simple Hypothesis experiment containing five stimuli (intro, three pairs of maps, and last slide) was used. Participants' task was to identify the differences between the maps. Coordinates of the clicks representing differences were also recorded.

OGAMA experiment was designed with only one screen recording stimulus. OGAMA in version 5.0 can record dynamic web stimuli, but it is not possible to use slides from Hypothesis as separate stimuli.

Recorded data were split according to their belonging to particular slides in the Hypothesis experiment. For the split, timestamps from Hypothesis indicating the slide change were used. The splitting and conversion of recorded data

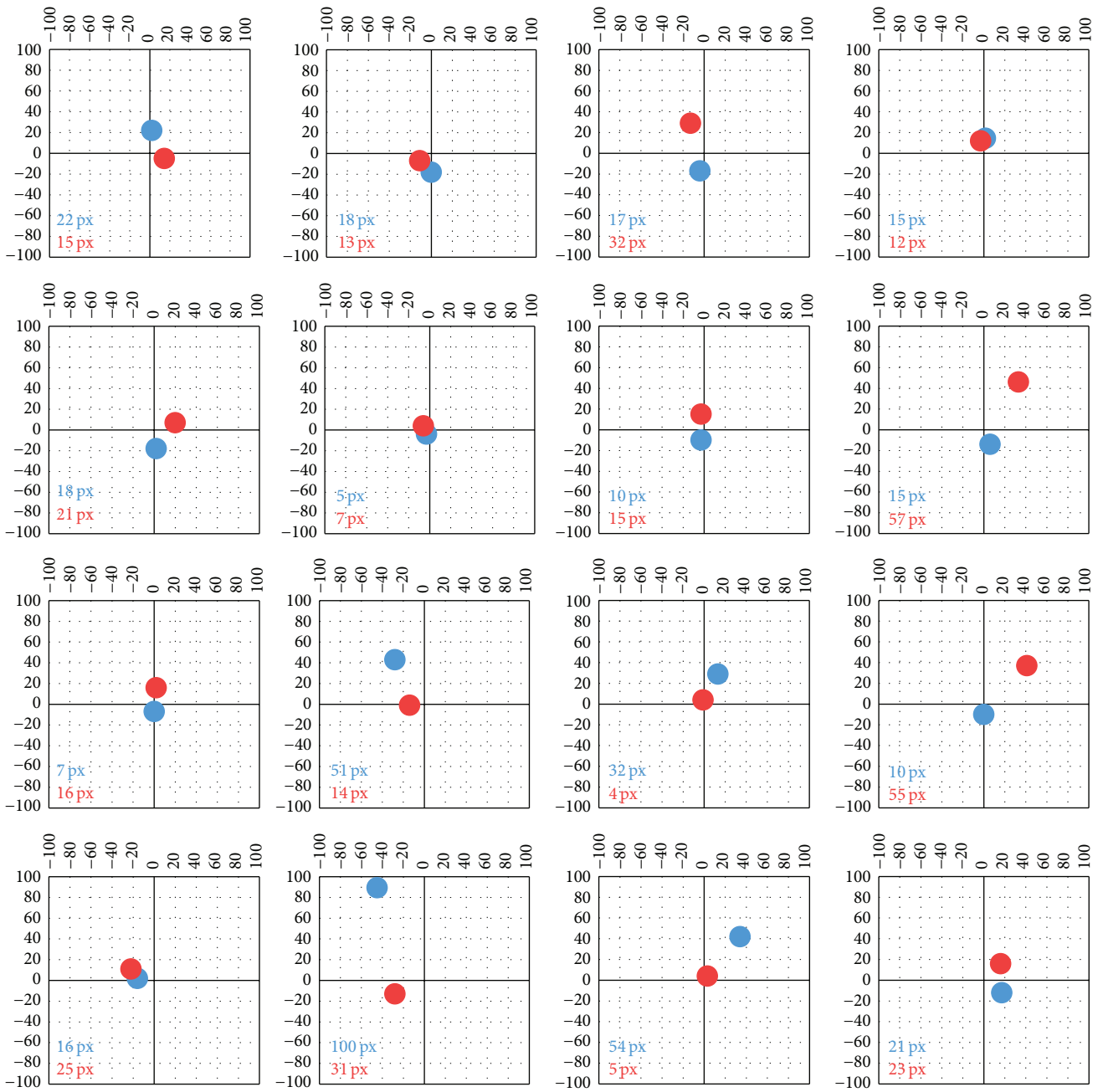


FIGURE 12: Comparison of fixation positions in Slide 2 for fourteen participants. Distance from the center of the image shows fixation deviation in pixels. EyeTribe data are displayed as blue dots; SMI data are displayed as red dots.

manually were time-consuming and not user-friendly. Thus, a web application called HypOgama was written in PHP for the automation of the process. The functionality of HypOgama application is illustrated in Figure 13.

The HypOgama application (Figure 14) is freely available at http://eyetracking.upol.cz/hypogama/.

The application synchronizes the Hypothesis time with the timestamp from the eye-tracking recording in OGAMA. The synchronization is processed by the key press that was used to start the Hypothesis experiment and which was recorded in both systems—in Hypothesis and OGAMA.

In the next step, the application scans the Hypothesis file and finds the timestamps of slide changes. These timestamps are then used for splitting raw eye-tracking data into blocks belonging to particular slides. The name of the relevant stimuli is added to all records from each block. In the final step, the data structure is modified for the direct import into a new OGAMA project.

The application contains six input fields:

(1) Exported file from Hypothesis manager containing data for one participant.

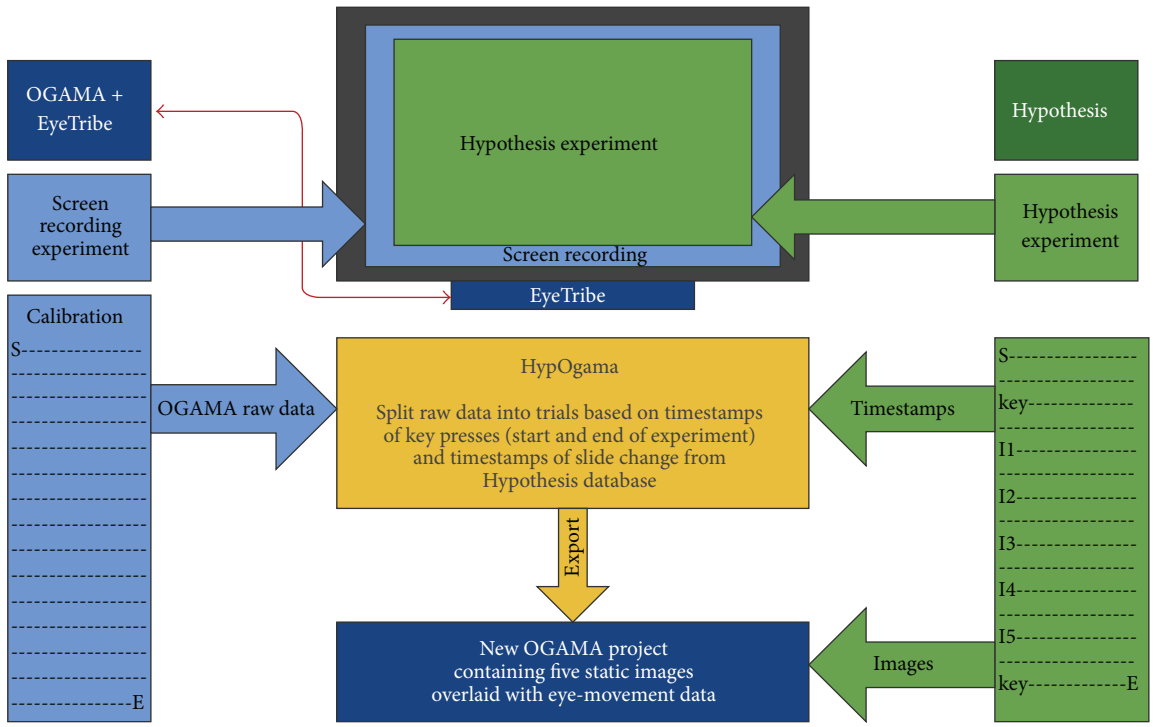


Figure 13: Process of splitting recorded data (screen recording) into trials with the use of HypOgama web application.

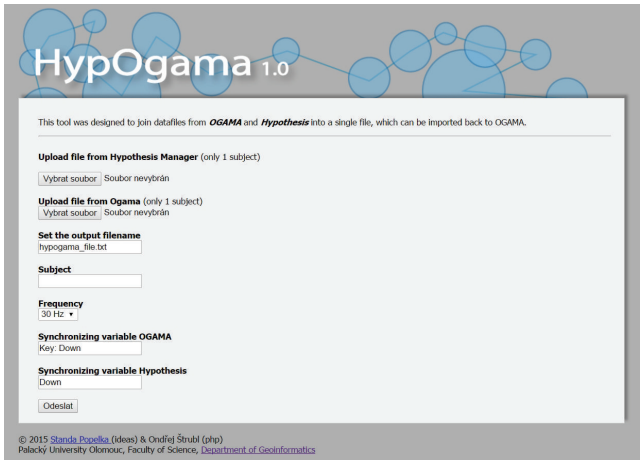


Figure 14: Environment of HypOgama web application.

(2) Exported raw data from the OGAMA application for one participant.

(3) Name of the output file.

(4) Subject name (if blank, the ID from Hypothesis will be used).

(5) Frequency of an eye-tracker (30 or 60 Hz).

(6) Synchronization variables: these values indicate which key was used for the synchronization of Hypothesis and OGAMA (default value is "Key: Down" in OGAMA format and "Down" in the format of Hypothesis application).

In the Hypothesis file (ad 1), HypOgama finds the row with the key press (default Key: Down) and the corresponding time, which corresponds to the beginning of the experiment. In the next step, the column containing the slide names is scanned and the time of the first occurrence of each slide is also stored. According to this time, OGAMA recording is split. The last information obtained from the Hypothesis file is the name of the subject, overwriting the subject name in the OGAMA file.

In OGAMA file, all records prior to the synchronization key press are erased. Stimuli names are replaced by those from Hypothesis file.

Outputs of the created script are raw eye-movement data for each slide that could be directly imported into the OGAMA project. The only one necessary thing is to put image files (stimuli) into OGAMA project folder. If it is the same filename as the one contained in the Hypothesis file, images will be automatically assigned to proper data. After the whole process, a user has OGAMA project containing static image stimuli with all corresponding eye and mouse movement data. The proposed concept was applied and verified through a selected case study described below. The purpose of this short study was to illustrate the functionality of interconnection of EyeTribe and OGAMA.

For the verification of the designed process of Hypothesis and EyeTribe combination, simple test battery was designed. For chosen procedure, Hypothesis was used for large-scale quantitative approach and eye-tracking method for the subsequent specification of certain results.

The test battery was established in the Hypothesis software and was focused on verification of Gestalt principles,

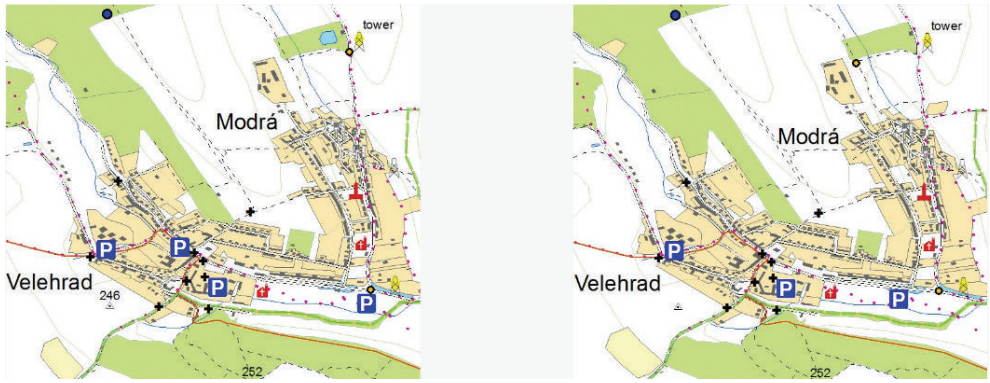


Figure 15: Example of stimuli—the first pair of topographic maps.

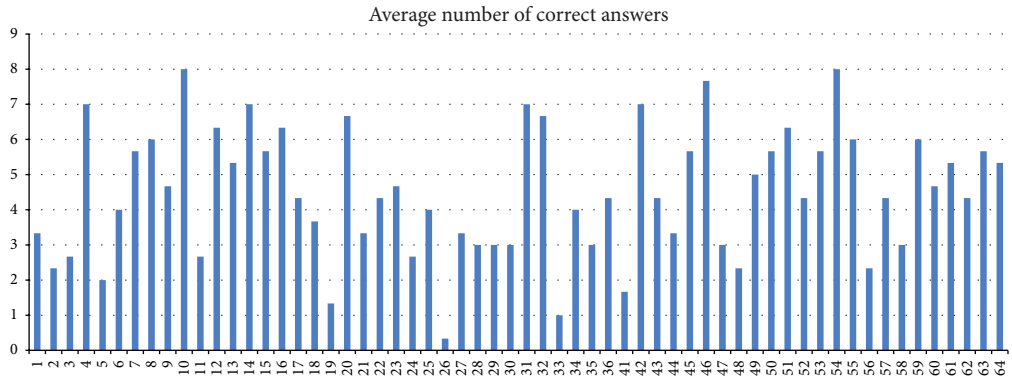


Figure 16: An example of results from Hypothesis. An average number of correct answers for each of the participants.

respectively, figure-ground organization, and on the cross-cultural comparison in the context of visual perception of cartographic stimuli [22, 32–35] on the example of specific cartographic products. The cartographic tasks were part of these more complex research batteries. The main purpose of this short cartographic study was the verification of HypOgama application and whole integrated research system for further research studies.

4.2. Participants. Participants of this illustrative case study were 64 students from the Masaryk University, Czech Republic, and 64 students from Wuhan University, China. In the first phase, participants were tested only on the web-based platform Hypothesis. Only a half of the dataset (Czech population) was further used in context of this particular study where the topographic and thematic maps were compared. In the second phase, the experiment was conducted with the use of eye-tracking system and the research sample is still continually extended.

4.3. Stimuli. The stimuli were represented by three pairs of maps that differed in 10 variables, for example, different colours of map signs, different position of the signs, and missing map signs. First two pairs of stimuli contained topographic maps. The third pair of the maps contained a thematic map.

The test was structured in three main parts. In the first part, participants filled out a personal questionnaire; in the second part, a representative example of the stimuli was presented to familiarize the participants with the environment of Hypothesis. In the third part, three tasks containing pairs of stimuli described above were presented. Participants were asked to mark the differences between presented maps. The time limit for each task was 45 seconds. An example of a topographic map (Slide 1) is displayed in Figure 15. On Slide 2, similar topographic map in different scale was shown. The last slide contained thematic map (see Figure 17).

4.4. Results and Discussion of Hypothesis and EyeTribe Interconnection. The performed study verified stability of proposed system on long distances and, at the same time, part of the test battery was used as a pilot study to verify the functionality of an integrated research system. Stimuli comparing the effectiveness of visual search between topographic and thematic maps were selected.

In the first phase, the test was performed in the Hypothesis application only. A number of differences identified between pairs of maps on Czech population were analysed (see Figure 16).

In the case of two pairs of topographic maps, the average number of correct answers was four. In the case of the stimuli

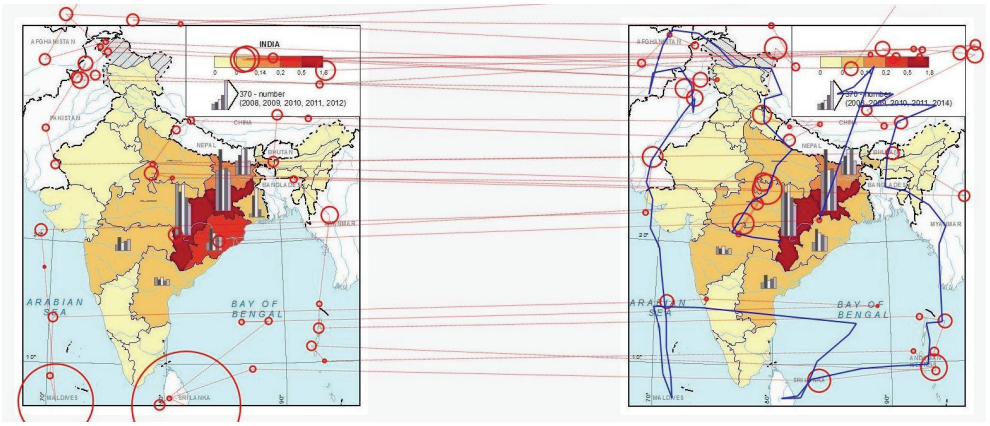


Figure 17: Example of eye-movement data recorded during the Hypothesis experiment. Circles represent fixations; blue line on the right is a mouse trajectory.

with a thematic map, the average number of correct answers was five.

To generalize the findings, an increase of the number of maps per condition would be necessary. However, this difference was the first clue to establish working hypotheses. Based on the data from the first phase of testing, hypotheses were established only at the level of stimulus-reaction. The way of task processing by users and their solving strategies were still a black-box; thus there was a need for more detailed procedural data, especially for information about distinct search strategies.

To explore differences in the visual search, eye-tracking can be used due to the ability to provide more detailed information (e.g., which kind of object was omitted, which kind of object could be found at first glance, and which areas attracts main attention).

Therefore, in the second phase, the already used experimental battery created in Hypothesis was interconnected with OGAMA through HypOgama application and the experiment was launched with the EyeTribe system. Cartographic stimuli and the eye-tracking data were linked together and further analysed with OGAMA.

The example in Figure 17 shows outputs from OGAMA-scan path and mouse trajectory of one participant over the stimulus with thematic maps. In this case, fixations are distributed mainly over the text labels in the map. Participant did not find the difference in the colour of the Odisha state (on the east coast of India) under the relatively large graph. At the same time, eye-tracking metrics (e.g., fixation count, dwell time for each map, and a number of saccades between these maps) can be statistically analysed. Based on findings from both types of analyses, the hypotheses for subsequent study can be established.

The functionality of the integrated research system has been fully verified in the above-mentioned pilot study. The experiment created on the Hypothesis platform was connected with OGAMA and EyeTribe via HypOgama. Data capture including eye-tracking recording continued and exploratory analyses of these data were performed.

5. Conclusion

The aim of the paper was to prove the concept of the mixed-research design through the interconnection of Hypothesis (software for experiment creation, experiment execution, and data collection) and the EyeTribe tracker (the most inexpensive commercial eye-tracker). This system could prove to be a valuable tool for cognitive cartography experiments and evaluation of user behaviour during map reading process.

The first necessary step was to evaluate the accuracy of the EyeTribe tracker with the use of concurrent recording together with the SMI RED 250 eye-tracker. The results of the comparison show that the EyeTribe tracker can be a valuable resource for cartographical research.

The next part of the study was focused on the interconnection of the EyeTribe with the Hypothesis platform, developed at Masaryk University in Brno. The connection was made through a newly created web application that modifies eye-movement data recorded during screen recording experiment in the OGAMA open-source application. The application is publicly available for the community of cartographers and psychologists at web page http://eyetracking.upol.cz/hypogama.

The interconnection advantages were illustrated on an example of simple case study containing three pairs of maps. The performed case study demonstrated the ability of the combined system of the Hypothesis platform and the EyeTribe tracker to support each other and to serve as an effective tool for cognitive studies in cartography.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This paper was supported by projects of Operational Program Education for Competitiveness (European Social Fund) (Projects CZ.1.07/2.3.00/20.0170 and CZ.1.07/2.3.00/30.0037)

of the Ministry of Education, Youth and Sports of the Czech Republic and the Student Project IGA_PrF_2015_012 of the Palacky University.

References

[1] G. L. Allen, C. R. Miller Cowan, and H. Power, "Acquiring information from simple weather maps: influences of domain-specific knowledge and general visual-spatial abilities," *Learning and Individual Differences*, vol. 16, no. 4, pp. 337–349, 2006.

[2] D. Edler, A.-K. Bestgen, L. Kuchinke, and F. Dickmann, "Grids in topographic maps reduce distortions in the recall of learned object locations," *PLoS ONE*, vol. 9, no. 5, Article ID e98148, 2014.

[3] P. Kubíček, Č. Šašinka, and Z. Stachoň, "Vybrané kognitivní aspekty vizualizace polohové nejistoty v geografických datech," *Geografie*, vol. 119, no. 1, pp. 67–90, 2014.

[4] E. S. Nelson, "Using selective attention theory to design bivariate point symbols," *Cartographic Perspectives*, vol. 32, pp. 6–28, 1999.

[5] E. S. Nelson, "Designing effective bivariate symbols: the influence of perceptual grouping processes," *Cartography and Geographic Information Science*, vol. 27, no. 4, pp. 261–278, 2000.

[6] A. H. Duc, P. Bays, and M. Husain, "Eye movements as a probe of attention," in *Progress in Brain Research*, C. Kennard and R. J. Leigh, Eds., vol. 171, chapter 5.5, pp. 403–411, Elsevier, 2008.

[7] N. Andrienko and G. Andrienko, *Exploratory Analysis of Spatial and Temporal Data. A Systematic Approach*, Springer, New York, NY, USA, 2005.

[8] J. M. Enoch, "Effect of the size of a complex display upon visual search," *Journal of the Optical Society of America*, vol. 49, no. 3, pp. 280–286, 1959.

[9] A. Hyrskykari, S. Ovaska, P. Majaranta, K.-J. Räihä, and M. Lehtinen, "Gaze path stimulation in retrospective think-aloud," *Journal of Eye Movement Research*, vol. 2, no. 4, pp. 1–18, 2008.

[10] Ö. Alaçam and M. Dalcı, "A usability study of WebMaps with eye tracking tool: the effects of iconic representation of information," in *Human-Computer Interaction. New Trends*, vol. 5610 of *Lecture Notes in Computer Science*, pp. 12–21, Springer, Berlin, Germany, 2009.

[11] S. I. Fabrikant, S. Rebich-Hespanha, N. Andrienko, G. Andrienko, and D. R. Montello, "Novel method to measure inference affordance in static small-multiple map displays representing dynamic processes," *Cartographic Journal*, vol. 45, no. 3, pp. 201–215, 2008.

[12] S. I. Fabrikant, S. R. Hespanha, and M. Hegarty, "Cognitively inspired and perceptually salient graphic displays for efficient spatial inference making," *Annals of the Association of American Geographers*, vol. 100, no. 1, pp. 13–29, 2010.

[13] K. Ooms, P. De Maeyer, and V. Fack, "Study of the attentive behavior of novice and expert map users using eye tracking," *Cartography and Geographic Information Science*, vol. 41, no. 1, pp. 37–54, 2014.

[14] S. Popelka and A. Brychtova, "Eye-tracking study on different perception of 2D and 3D terrain visualisation," *Cartographic Journal*, vol. 50, no. 3, pp. 240–246, 2013.

[15] J. M. Olson, "Cognitive cartographic experimentation," *Cartographica*, vol. 16, no. 1, pp. 34–44, 1979.

[16] Z. Štěrba, Č. Šašinka, Z. Stachoň, P. Kubíček, and S. Tamm, "Mixed research design in cartography: a combination of qualitative and quantitative approaches," *Kartographische Nachrichten*, vol. 64, no. 5, pp. 262–269, 2014.

[17] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, Sage, Thousand Oaks, Calif, USA, 2nd edition, 2003.

[18] Z. Štěrba, Č. Šašinka, Z. Stachoň, K. Morong, and R. Štampach, *Selected Issues of Experimental Testing in Cartography*, Masaryk University, 2015, http://munispace.muni.cz/index.php.

[19] M. Konečný, Š. Březinová, M. V. Drápela et al., *Dynamická Geovizualizace v Krizovém Managementu*, Masarykova Univerzita, Brno, Czech Republic, 2011 (Czech).

[20] P. Kubíček, Č. Šašinka, and Z. Stachoň, "Uncertainty visualisation testing," in *Proceedings of the 4th Conference on Cartography and GIS*, Sofia, Bulgaria, June 2012.

[21] Z. Stachoň, Č. Šašinka, P. Kubíček, and Z. Štěrba, "MuTeP—alternativní nástroj pro testování kartografických vizualizací a sběr dat," in *Proceedings of the 23rd Sjezd České Geografické Společnosti*, Prague, Czech Republic, 2014 (Czech).

[22] Z. Stachoň, Č. Šašinka, Z. Štěrba, J. Zbořil, Š. Březinová, and J. Švancara, "Influence of graphic design of cartographic symbols on perception structure," *Kartographische Nachrichten*, vol. 63, no. 4, pp. 216–220, 2013.

[23] K. Morong and Č. Šašinka, "Hypothesis—online software platform for objective experimental testing," in *Proceedings of the Applying Principles of Cognitive Psychology in Practice*, Brno, Czech Republic, May 2014.

[24] S. W. T. Widgets, https://www.eclipse.org/swt/.

[25] E. Dalmaijer, "Is the low-cost EyeTribe eye tracker any good for research?" *PeerJ PrePrints*, vol. 2, Article ID e585v1, 2014.

[26] K. Ooms, L. Dupont, L. Lapon, and S. Popelka, "Accuracy and precision of fixation locations recorded with the low-cost Eye Tribe tracker in different experimental set-ups," *Journal of Eye Movement Research*, vol. 8, no. 1, pp. 1–24, 2015.

[27] A. Voßkühler, V. Nordmeier, L. Kuchinke, and A. M. Jacobs, "OGAMA (Open Gaze and Mouse Analyzer): open-source software designed to analyze eye and mouse movements in slideshow study designs," *Behavior Research Methods*, vol. 40, no. 4, pp. 1150–1162, 2008.

[28] S. Popelka, "Optimal eye fixation detection settings for cartographic purposes," in *Proceedings of the 14th SGEM GeoConference on Informatics, Geoinformatics and Remote Sensing (SGEM '14)*, vol. 1, pp. 705–712, June 2014.

[29] S. Popelka and J. Doležalová, "Non-photorealistic 3D visualization in city maps: an eye-tracking study," in *Modern Trends in Cartography*, pp. 357–367, Springer, 2015.

[30] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye Tracking: A Comprehensive Guide to Methods and Measures*, Oxford University Press, Oxford, UK, 2011.

[31] J. Nielsen, *Why You Only Need to Test with 5 Users*, Nielsen Norman Group, 2000, http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/.

[32] J. Čenek, "Individualism and collectivism and their cognitive correlates in cross-cultural research," *The Journal of Education, Culture and Society*, vol. 2, pp. 210–225, 2015.

[33] R. E. Nisbett, I. Choi, K. Peng, and A. Norenzayan, "Culture and systems of thought: holistic versus analytic cognition," *Psychological Review*, vol. 108, no. 2, pp. 291–310, 2001.

[34] E. Rubin, "Figure and ground," in *Visual Perception*, S. Yantis, Ed., pp. 225–229, Psychology Press, Philadelphia, Pa, USA, 2001.

[35] J. Wagemans, J. H. Elder, M. Kubovy et al., "A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization," *Psychological Bulletin*, vol. 138, no. 6, pp. 1172–1217, 2012.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 8680541, 14 pages
http://dx.doi.org/10.1155/2016/8680541

Review Article

Low Cost Eye Tracking: The Current Panorama

Onur Ferhat[1,2] and Fernando Vilariño[1,2]

[1]*Computer Vision Center, Edifici O, Campus UAB, 08193 Bellaterra, Spain*
[2]*Computer Science Department, Universitat Autònoma de Barcelona, Edifici Q, Campus UAB, 08193 Bellaterra, Spain*

Correspondence should be addressed to Onur Ferhat; oferhat@cvc.uab.es

Received 27 November 2015; Accepted 18 February 2016

Academic Editor: Ying Wei

Despite the availability of accurate, commercial gaze tracker devices working with infrared (IR) technology, visible light gaze tracking constitutes an interesting alternative by allowing scalability and removing hardware requirements. Over the last years, this field has seen examples of research showing performance comparable to the IR alternatives. In this work, we survey the previous work on remote, visible light gaze trackers and analyze the explored techniques from various perspectives such as calibration strategies, head pose invariance, and gaze estimation techniques. We also provide information on related aspects of research such as public datasets to test against, open source projects to build upon, and gaze tracking services to directly use in applications. With all this information, we aim to provide the contemporary and future researchers with a map detailing previously explored ideas and the required tools.

1. Introduction

From a computer scientist's perspective, human beings are machines which receive input from their sensors such as ears, eyes, and skin and which interact with the world they live in through their actuators, which are their hands, feet, and so on. Their attention can be understood by analyzing the way they direct their sensors (i.e., looking at specific locations or inspecting unknown objects by touching or smelling). Moreover, as in the case of robots, examining this attention can give us hints about their state of mind and their way of reasoning.

Among the human senses, sight has an important place in today's world where we are surrounded with digital displays be it in our mobile phones, our computers, or televisions. Instead of making passive observations of the objects around, it also gives hints about what the person actively chooses to see through eye movements. Analysis of these movements, therefore, sparked great interest in research communities.

Devices or systems that track a person's eye movements are called eye trackers or gaze trackers. Currently, the most widespread techniques used in these trackers make use of light sources and cameras that operate in the infrared (IR) spectrum. There are many available commercial models that are in the form of either eyeglasses or table mounted devices [1–3] and also open source alternatives that allow the use of custom hardware [4].

Visible light gaze tracking, on the other hand, does not require any special hardware and aims to solve the task making use of regular cameras. In this paper, we will concentrate on this class of trackers and survey the related research. Furthermore, we will limit our search to the table mounted setup (also called remote setup) as it is ubiquitous in contemporary devices and it removes the restrictions for camera placement (with a few exceptions). Our aim and contribution is as follows:

(i) To provide an exhaustive literature review.

(ii) To comment on these works from various perspectives.

(iii) To list publicly available datasets.

(iv) To list open source software.

(v) To list gaze trackers as a web service.

The rest of the paper is organized as follows: we will start with an overview of the software structure used in remote, visible light gaze trackers. Then, we will categorize and explain the previous work according to the techniques

used and continue with two other categorization schemes: how/if they are calibrated and how/if they handle head movements. Afterwards, we will list and comment on the available datasets, online gaze tracking services, and open source projects. We will finish with our conclusions regarding the current state and future directions.

2. Categorization and Structure of Visible Light Gaze Trackers

The categorization of the works that we analyze in this paper is not trivial, because the borders between groups of methods are not always clear and in the literature different naming schemes exist.

In the early review by Morimoto and Mimica [5], methods using the eye appearance (i.e., eye region image pixels) directly for gaze estimation are called appearance-based or view-based methods, and the rest is left unnamed. Here, the given name refers to all the visible light methods and does not give information about the subcategories. Even in a more recent survey [6] where both infrared (IR) and visible light methods are considered, the latter group is considered as just an alternative, and its subcategories are left unclear. Other categorization schemes also build on this ambiguity: appearance-based versus feature-based [7, 8] and appearance-based versus model-based [9, 10]. It should also be noted that the "appearance-based" name is still being used to refer to all visible light methods [11, 12], adding to the confusion.

With the aim of clearly identifying the borders between different visible light gaze estimation techniques (and hopefully not adding to the confusion), we present a new categorization scheme:

(1) *Appearance-Based*. These methods only use the eye image pixel intensities to create a mapping to the gaze estimation. The image pixels are converted to a vector representation via raster scanning and fed to the estimation component.

(2) *Feature-Based*. Methods of this category also make use of a mapping to calculate the gaze; however, they use richer feature vectors compared to the methods in the previous category (i.e., not just pixel intensities).

(3) *Model-Based*. Compared to the discriminative approach of the first two categories, the methods belonging to this category follow a generative approach by trying to model the eyes and maybe even the face. The gaze is calculated geometrically using the model parameters.

After explaining our categorization and the reasoning behind it, we can continue with the discussion about the software pipeline of these trackers. Although the variation in details is huge, a common skeletal structure that describes their software implementation can easily be identified as seen in Figure 1.

The input to the system is generally a video stream; however, examples of systems working on still images are also found [13]. In the former case, the previously processed video

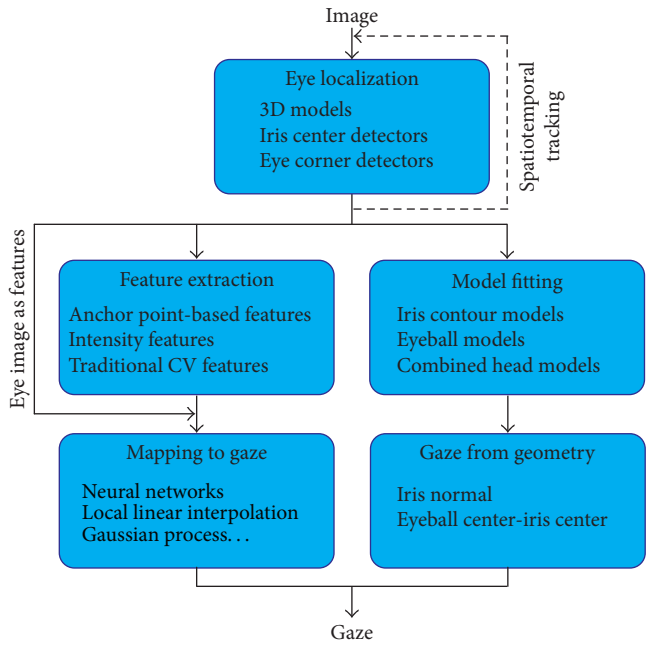


Figure 1: The common software structure for visible light gaze trackers. The methods start by locating the eyes. To make the estimation more stable, spatiotemporal tracking may be utilized at this step. Later, the location information is used to extract features, fit 2D or 3D models to the eyes, or just extract the eye region image. In the case of model-based methods, the fitted model is used to calculate the gaze geometrically, whereas in the other methods, a mapping function is necessary to calculate the gaze angle or point.

frames' results may be used to improve the performance for the next frames [14].

The first task in the pipeline is to extract the eye region. If an optional head pose estimation component is present, and if its output contains information about the eye location, it may be used directly as the location or it may be used as a rough initial estimate for the actual eye locator. Otherwise, the eye locator component has the option of using face detectors to restrict the processed image area and reduce computational cost [15, 16]. In order to calculate accurate eye location, the system can make use of iris center detectors [17], eye corner detectors [18], or 3D eye models that take into account the appearance of the entire eye [19].

Once the region of interest (ROI), that is, the eye region, is located, the second step is to prepare the input for the gaze estimation component. Depending on the class of gaze estimation method, the required input for the last step varies. In *appearance-based* methods, the extracted eye image from the first step is used directly as the input. Here, each image pixel intensity is considered as one dimension of the input vector. As the change in illumination and shadows may interfere with these inputs, this class of methods may not always give robust results.

Feature-based methods try to break the direct connection between the raw pixel intensities and the final input vector, in an attempt to increase robustness to lighting changes. Some of the features used in the literature are as follows:

(i) Pixel positions of keypoints (e.g., inner eye corners, iris center, and eyelid) [20, 21].

(ii) Their relative positions (i.e., vectors connecting two positions) [22–24].

(iii) Standard computer vision features such as histogram of oriented gradients (HOG) [25, 26] and local binary patterns (LBPs) [11, 27].

(iv) Features calculated by a convolutional neural network (CNN) [13].

(v) Features grouping and summarizing pixel intensities [28–31].

Finally, the *model-based* gaze estimation methods require the parameters for a 2D or 3D eye model as the input. In case of 2D, these can be the parameters defining the iris edge model [32]; in the 3D case, it can get more complex to include 3D positions of the eyeball center [33] or other facial landmarks [34].

The last step in the described pipeline is the estimation of the gaze, given the inputs calculated in the previous step. Appearance-based and feature-based methods require a mapping function that maps the input vectors to the gaze point or the gaze direction. The commonly used techniques include neural networks (NNs) [35, 36], Gaussian process (GP) regression [14, 37], and linear interpolation [38, 39]. On the other hand, model-based methods use the geometry of their 3D model (e.g., normal vector for the iris of 3D eye ball model) to calculate the gaze [40, 41].

3. Methods for Single Camera Remote Gaze Tracking

In this section, we categorize the works that we focused on according to our scheme. A summary of these works can be seen in Table 1.

3.1. Appearance-Based Methods. The first techniques proposed for visible light gaze tracking introduced the category of appearance-based methods [16, 35, 42]. These methods are characterized by their use of eye image pixel intensities as their features for gaze estimations. After a possible histogram normalization step for standardizing image appearances over the whole dataset, these feature vectors are fed to the estimation component which maps them to screen coordinates.

3.1.1. Neural Networks. One of the most popular mapping functions used in eye tracking is neural networks (NNs). In their pioneering work, Baluja and Pomerleau [35] introduce the first method making use of NNs. They test their system extensively by varying the inputs (iris region or entire eye), NN structure (single continuous or divided hidden layer), and the hidden layer unit number. In another experiment, they demonstrate that, by training the system with inputs from different head poses, the system can even handle small head movements. Finally, they top their system with an offset table that is used to correct the systematic shifts in actual eye tracker use. In the best case, their reported accuracy is around 1.5°.

Stiefelhagen et al. [16] use skin color segmentation and pupil detection to replace the use of a light source for this task in the original work of Baluja and Pomerleau. Xu et al. [42] introduced an iterative thresholding method to locate the iris region accurately and also proposed Gaussian smoothing for outputs of the NN during training. Two recent works [43, 44] used the NN technique for gaze tracking on commercial tablet computers and report lower accuracy (average error > 3°), mainly because of the low sampling rates in tablets and high training data demand of the NNs.

3.1.2. Local Linear Interpolation. A recently more popular alternative to NN mapping is local linear interpolation as proposed for gaze tracking by Tan et al. [38]. In their work, they see the eye region images as coming from an appearance manifold, and gaze estimation is posed as a linear interpolation problem using the most similar samples from this manifold. Although this work makes use of IR illumination for eye localization, the gaze estimation technique is valid for purely visible light setups. The reported accuracy of around 0.40° shows the promise of the proposed technique.

Ono et al. [45] calculate the decomposition of the eye image, which takes into account variations caused by gaze direction, base eye appearance, and shifts in image cropping. Using this decomposition, they can encounter the most similar 3 training samples and use LLI to calculate the gaze with 2.4° accuracy.

Sugano et al. [46] use an LLI technique that allows head movements. They cluster the eye images according to the corresponding head pose and choose samples for interpolation only from the cluster with the same head pose as the current sample. Their system keeps learning from user interaction (i.e., mouse clicks) and continuously updates its parameters, adding clusters for new head poses when necessary. The reported average error is in the range 4-5°. The extended version of the work [47] provides methods for refining gaze labels acquired through mouse clicks, discarding high-error training samples, and locating the eye position better, thus decreasing the average error to only 2.9°.

Lu et al. [7, 29] decompose the gaze estimation problem into subproblems: (1) estimation under fixed head pose and (2) compensation of errors caused by head rotation and eye appearance distortion. Unlike other work, they do not choose the most similar local training samples explicitly; however, they argue that their method for weighting all the training samples automatically selects a small number of local samples. By learning eye appearance distortion from 5-second video clips and applying both compensations, they decrease the average error from 6° to 2.38° (and from 13.72° to 2.11° in the 2014 paper). In their later work [48, 49], instead of video clips (containing around 100 frames), they acquire only 4 additional training samples under reference head poses and synthesize extra training samples by modeling the change in eye appearance.

Alnajar et al. [50] propose a calibration-free estimation based on the assumption that humans have similar gaze patterns for the same stimulus. Here, first initial gaze points are calculated for a user without calibration, and then a transformation is calculated to map the user's gaze pattern

Table 1: Summary and results of all the techniques analyzed in this work. Methods are grouped into categories for easier reference. HP column shows whether the technique has head pose invariance or not. Techniques allowing small head movements are denoted by ≈ symbol. Output column shows what type of gaze is calculated: point of gaze (◦) or line of gaze (∠).

	Feature	Mapping	Calibration	HP	Dataset	Output	Accuracy	References	Comments
Appearance-based	—	NN	Grid	—	—	◦	1.5–4	[16, 35, 42–44]	
	—	GP	Grid	—	[98]	◦	2	[9]	
	—	GP	Grid	≈	—	◦	n/a	[37, 53]	Rigorous calib. for HP
	—	LLI	Grid	—	—	◦	0.4	[38]	IR to locate eye
	—	LLI	Grid	—	—	◦	2.4	[45]	
	—	LLI	Grid + HP	√	—	◦	2.2–2.5	[7, 29, 48, 49]	0.85° error with fixed HP
	—	LLI	Grid	√	—	∠	4.8	[8]	
	—	LLI	—	√	—	◦	3–5	[46, 47]	Incremental calibration
	—	LLI	Grid	√	[99]	∠	4	[51]	8 cameras
	—	LLI	—	—	—	◦	3.5–4.3	[10, 50]	Saliency for calibration
Feature-based	PC-EC	GP	Grid	—	—	◦	1.6	[20, 54]	
	PC-EC	LI	Grid	—	—	∠	1.2	[22]	
	PC-EC	LI	Grid	—	—	◦	n/a	[24, 55]	
	PC-EC	PI	Grid	—	—	◦	1.2	[39]	3° without chin rest
	PC-EC	LI	Grid	√	—	◦	2–5	[56]	
	PC-EC	PI	Grid	—	—	◦	2.4	[57]	
	PC-EC	PI	Grid	√	—	◦	2.3	[18]	1.2° error with fixed HP
	Several	NN	Grid	—	—	◦	1-2	[23, 58]	
	Several	NN	Grid	√	—	◦	2–7	[21]	Few tests
	EC shift	n/a	Grid	—	—	◦	3.2	[59]	
	EC shift	LI	—	—	—	◦	3.4	[60]	Hand-coded params.
	GC-CM	LI	Grid	—	—	◦	1.5	[62]	
	Several	LI	Grid	—	—	◦	3	[17]	
	Edge energy	S^3GP	Grid	—	—	◦	0.8	[14]	
	Intensity	ALR	Grid	≈	—	◦	0.6	[28, 63]	8D or 15D feats.
	Intensity	RR	Grid	—	—	◦	1.1	[31]	120D feats.
	HOG	SVR/RVR	Grid	—	—	◦	2.2	[26]	
	Several	NN	Grid	—	—	◦	3.7	[36]	Dim. reduced to 50
	CS-LBP	S^3GP	Grid	—	—	◦	0.9	[11]	Partially labelled data
	Several	Several	Grid	—	[100]	◦	2.7	[66]	Dim. reduced to 16
	Several	Several	Grid	√	[101]	◦	3.2	[67]	
	CNN	Several	Continuous	√	[68]	∠	~6	[13]	Calib. from dataset
	Segmentation	GP	Grid	—	—	◦	2.2	[30]	
	Model		Calibration	HP	Dataset	Output	Accuracy	References	Comments
Model-based	Iris contour		Camera	√	—	∠	1	[32, 70]	One-circle alg.
	Iris contour		Grid	√	—	◦	4	[71, 72]	
	Iris contour		—	√	—	∠	n/a	[73]	Two-circle alg.
	Iris contour		Camera	—	—	◦	n/a	[74]	
	Iris contour		Camera	—	—	∠	0.8	[75]	Error for single dir.
	Iris contour		Grid	√	—	∠	3.3	[76]	
	Iris contour		Grid	√	—	∠	3.5	[77]	
	Iris contour		Grid	√	—	◦	6.9	[12]	
	Eyeball		Grid	√	—	∠	3.2	[34]	Calib. personal params.
	Eyeball		Grid	—	—	∠	3.5	[40]	PF tracking
	Eyeball		1 target	√	—	∠	~2	[78]	Error for single dir.
	Eyeball		Grid	√	—	◦	2.7	[33]	
	Eyeball		—	√	—	∠	9	[19, 79]	Autocalibration
	Eyeball		Grid	√	—	◦	n/a	[41]	
	Eyeball		—	√	[102, 103]	∠	5.6	[80]	

to other users. For the initial gaze estimation, they either use the closest neighbors from the training set to reconstruct the current eye appearance (with samples from other users) or project the eye appearance to a 2D manifold to get the most similar samples.

Lai et al. [8] use random forests to learn the neighborhood structure for their joint head pose and eye appearance feature (HPEA). Gaze is estimated with linear interpolation using the neighbors in the random forest, yielding an accuracy of around 4.8° (horizontal and vertical combined).

Sugano et al. [51] build a multiview dataset and use it to reconstruct part of the face in 3D. They use this 3D model to generate synthetic samples acquired from different camera angles and use the extended dataset to train a random forest. Here, unlike their previous work [46], they do not divide the data strictly according to the head pose; however, they build sets of regression trees with overlapping head pose ranges (i.e., samples from a single head pose are used in building several sets of trees). Gaze is calculated as the average result from the nearest regression forests according to head pose, resulting in an average error of 6.5° with cross-subject training.

3.1.3. Gaussian Processes. Gaussian process (GP) is another choice for the mapping in some gaze tracking methods. GP predictions are probabilistic and allow calculation of confidence intervals for the outputs which may be used as an indicator to detect when the calibration is no longer valid for the test data [20, 52].

Nguyen et al. [37, 53] describe a system where they use a Viola and Jones [15] eye detector and optical flow (OF) to detect and track the eye in the camera image. Then, the extracted eye image is fed to a GP to calculate the gaze point. In the extended work [37], they show that when the calibration is repeated in several head poses, the system can even become head pose invariant.

Ferhat et al. [9] also propose a similar method, where they use several Viola-Jones detectors (face, eye, nose, and mouth) to choose 8 anchor points on the face automatically and use the extracted eye image to train a GP. In the final system, the average error is 2° (horizontal and vertical combined).

Sugano et al. [10] use saliency information to automatically calibrate a gaze tracker while the subject is watching a video clip. While calibrating the GP-based tracker, instead of using known gaze positions, they train the GP with gaze probability maps calculated by aggregating several saliency maps.

3.2. Feature-Based Methods. In the appearance-based methods, the inputs to the mapping functions were the same across all techniques; therefore, we categorized them according to the mapping functions they used. However, in feature-based methods, the main difference is their feature set, and our categorization also reflects this difference.

3.2.1. Anchor Point Position-Based Features. In this first subcategory of feature-based methods, the positions of important anchor points inside and around the eye (e.g., pupil (iris) center, inner and outer eye corners, and nostrils) are used as features. In some cases, they constitute distinct dimensions of the feature set, whereas in other cases, the relation between them (i.e., the vector connecting two anchor points) is used as the feature.

Pupil Center-Eye Corner Vector. In infrared gaze trackers, a feature widely used for gaze estimation is the pupil center-corneal reflection vector (PC-CR) [39]. The equivalent of this in natural light methods is the pupil center-eye corner vector (PC-EC) (or, alternatively, iris center-eye corner (IC-EC) vector).

The first use of the PC-EC vector in natural light eye trackers is proposed by two distinct research groups around the same time [20, 22, 54]. Hansen et al. [20, 54] use Active Appearance Model (AAM) and mean shift to track the eyes over time and find the positions of pupil center and eye corners. Gaze estimation is done by training a Gaussian process (GP) where the input is the PC-EC vector. The system results in an average error of around 1.6°, and the eye tracker is verified in an eye-typing interface. Zhu and Yang [22], on the other hand, propose methods for detecting the iris center and the eye corner with subpixel accuracy. They use a 2D linear mapping to estimate gaze positions from the feature vectors. They report an accuracy of around 1.2° from their experiments.

Valenti et al. [24, 55] propose a novel eye corner locator and combine it with a state-of-the-art eye center locator to calculate the EC-PC vector. Inspired by Zhu and Yang [22], they also use a 2D linear mapping for gaze estimation. In their later work [56], they make use of a head pose estimator and use the calculated transformation matrix to normalize the eye regions. The more accurate eye location found this way, in turn, is used to better estimate the head pose in a feedback loop. To solve the gaze estimation problem with head movements, they *retarget* the known calibration points to the monitor coordinates whenever there is a change in the head pose and calibrate the system again. With these improvements, they achieve average errors of between 2° and 5° in two experimental tasks.

Sesma et al. [39] normalize the PC-EC vector, dividing the vector components by the Euclidean distance between the inner and outer eye corners. For gaze estimation, they use both PC-EC vectors for the inner and outer eye corners and their experiments show the average error to be 1.25° when the head movement is constrained and around 3° when no chin rest is used.

Baek et al. [57] apply image rectification to rectify the eye images to a front facing head pose and combine it with a novel iris center localization method. They use second-order polynomial equations (as in [39]) to calculate the gaze and measure an accuracy of 2.42°.

Cheung and Peng [18] fit Active Shape Models (ASM) on images normalized using local sensitive histograms. With the novel methods they propose for iris center and eye corner detection, they achieve errors of 1.28° with fixed head pose and 2.27° with head movements.

Others. Some feature-based methods making use of anchor point positions may take a different path and combine or replace the EC and PC positions with information coming

from other anchor points (e.g., nostrils) or simply calculate the features in another way.

In his thesis, Bäck [21] uses several geometrical features such as iris center, eye corner, nostril positions, head angle, and eye angles to create a rich feature vector and trains a NN for gaze estimation. The system is not tested heavily; however, the accuracy is reported to be in the range 2–4° and sometimes even up to 7-8°.

Torricelli et al. [23, 58] calculate several distance and angle features from both eyes to fill the feature vector. These features include distances of inner and outer eye corners to the iris center, the slopes of the lines connecting these points, and the positions of outer eye corners. The trained NN gaze estimation component results in average errors in the range 1-2°.

Ince and Kim [59] track the iris with a custom method and calculate the gaze using the iris center displacement between subsequent camera frames. The proposed system has an accuracy of 3.23° (horizontal and vertical combined). Nguyen et al. [60] take a similar approach and make use of the center-bias effect, which states that gaze distribution is biased towards the center of the screen [61]. Their system does not require any calibration and works by calculating the mean iris center over time and estimating the gaze through the difference of current iris center and the mean. The combined error in x and y directions is 3.43° of visual angle.

Wojciechowski and Fornalczyk [62] preprocess the eye images by calculating the edges and then extract their features which are the geometric center and center of mass of edge pixel positions. The final feature is the vector connecting these two locations (GC-CM), which is used to calculate the gaze estimation using the weighted average of data from 4 training points. The system has around 1.5° accuracy (combined).

Skodras et al. [17] track several moving and stationary anchor points (e.g., eye corner, eyelid control points, and iris center) and calculate vectors from their relative positions to build the final feature vector. They use linear regression for mapping to gaze point and achieve an accuracy of 2.96° (combined).

3.2.2. Intensity-Based Features. In some feature-based methods, the direct connection between the image pixel intensity and feature vector is not broken completely. Williams et al. [14] combine the image pixel intensities with edge energies in their feature vector. They train a sparse, semisupervised Gaussian process (S^3GP) which also infers the missing labels in the partially labeled training data. They make use of the confidence value for the GP to filter the estimation over time using a Kalman filter and achieve a final accuracy of 0.83°.

Lu et al. [28, 63] propose extracting 8D or 15D intensity features from the eye region, which is identical to resizing the grayscale eye image to 2×4 or 3×5 pixels, respectively. Together with the proposed subpixel alignment method for eye region, and adaptive linear regression (ALR) for gaze estimation, they can estimate the gaze point with up to 0.62° accuracy.

Xu et al. [31] extend the work of Lu et al. [28, 63] to increase the feature dimension to 120D (2 eye images of 6×10 pixels) and to use ridge regression for gaze estimation and achieve slightly worse results (1.06°).

3.2.3. Traditional Computer Vision Features. Computer vision (CV) tasks such as object detection and classification are normally solved by using features (e.g., histogram of oriented gradients (HOG) [25], scale-invariant feature transform (SIFT) [64], and local binary patterns (LBPs) [27]) extracted around salient points in the images. However, until recently, this approach was still unexplored for the gaze tracking problem.

Martinez et al. [26] introduce this concept in a head mounted tracker, where they extract multilevel HOG features from eye images and use support vector regression (SVR) or relevance vector regression (RVR) to map these features to the gaze point, and achieve an accuracy of 2.20° (combined).

Zhang et al. [36] combine several features to build their feature vectors: color, pixel intensity, orientation (from several Gabor filters), Haar-like features, and spatiogram features (combining color histogram with spatial information). After generating this rich representation, they apply a dimensionality reduction technique to reduce the feature vector size to 50 and train a NN for gaze estimation. Although the reported average error is not very low (around 3.70°, when combined), the work is a great example of applying the traditional CV pipeline to gaze trackers.

Liang et al. [11] build on the previously explained S^3GP technique [14] and train it with CS-LBP features [65], which is based on LBPs. They make use of spectral clustering to learn from partially labeled data and report an average error of 0.92°.

Schneider et al. [66] explore several feature types (DCT, LBP, and HOG) in conjunction with many alternatives for regression (GP, k-nearest neighbors (kNN), regression trees, SVR, RVR, and splines). They use a *dually supervised embedding* method to reduce the feature dimensionality, resulting in up to 31.2% decrease in the errors (best accuracy being 2.69° with 16-dimensional features based on HOG and LBP). Huang et al. [67] also take the same approach and review several feature types (LOG, LBP, HOG, and mHOG) and regression components (kNN, RF, GP, and SVR). They report that random forests (RF) combined with multilevel HOG (mHOG) features prove to be the most efficient combination (3.17° error) in a very challenging scenario (i.e., tablet computers), with free head movements.

Lately, convolutional neural networks (CNNs) are very popular in computer vision research, and Zhang et al. [13] are the first to use them for gaze tracking. CNN methods generally require a large dataset, and in their work they also present their dataset [68] which contains more than 200,000 images. They calculate features using a CNN and combine these features with the head pose information to build the complete feature vector. After testing with several regression functions (random forests, kNN, ALR, and SVR), the best accuracy they achieve is around 6°.

3.2.4. Others. Ferhat et al. [30] use the segmented iris area to calculate their proposed features. In their feature vector (which contains 192 dimensions for an eye image of size

128 × 64), a given feature dimension holds the number of segmented pixels in the corresponding row or column of the iris segmentation image. Their system makes use of GP for regression and has an accuracy of 2.23° (combined).

3.3. Model-Based Methods. The models used in model-based gaze estimation methods are roughly divided into two: iris contour models (also known as one-circle algorithm), where an ellipse is fitted around the iris region, and eyeball models, where the main objective is to estimate the location of the eyeball center.

3.3.1. Iris Contour Models. The direct least squares method for fitting ellipses onto a set of points [69] is influential in the development of iris contour models for gaze estimation. This method, complemented with the observation that the circular iris boundary appears as an ellipse in camera images, has enabled the development of several gaze tracking techniques.

Wang et al. [32, 70] develop the one-circle algorithm where they use edge detection to find pixels belonging to the iris boundary, and they fit an ellipse to this set of locations. Then, the ellipse is back-projected to the 3D space to find the iris contour circle, and its normal vector is used as the gaze vector. Their system has an average error of around 1°.

Hansen and Pece [71, 72] use an active contour method to track the iris edges over time, and (probably) using the one-circle method, their system estimates the gaze with around 4° accuracy.

Wu et al. [73] propose an extension with their two-circle algorithm, where they assume the elliptic iris contours for both eyes lie on the same plane or on parallel planes in 3D. With this assumption, they are able to estimate the gaze direction without the need for camera calibration.

Huang et al. [74] use randomized Hough transformation for iris contour fitting, whereas Zhang et al. [75] propose an improved RANSAC algorithm. The reported that accuracy for the latter work is 0.8° in a single direction.

Fukuda et al. [76] propose subpixel methods for iris contour estimation in low resolution images, achieving a combined average error of 3.35°. Mohammadi and Raie [77] train a support vector machine (SVM) to filter out the unrelated edge segments before applying the ellipse fitting, yielding an accuracy of 3.48°.

Wood and Bulling [12] detect the edges belonging to the iris from the image's radial derivative. After fitting the ellipse using the RANSAC method, the gaze estimation has an accuracy of 6.88°.

3.3.2. Eyeball Models. Eyeball model-based techniques try to infer the eyeball center position and calculate the gaze vector as the line connecting this point with the iris center.

Ishikawa et al. [34] use an AAM to track the face and use the eye corner positions and the scale of the face to infer the anatomical constants for the user (i.e., eye geometry). This calibration is followed by iris detection by template matching and edge-based iris refinement to calculate the center of the iris. The geometrically calculated gaze has an average error of 3.2°.

Wu et al. [40] track the iris contours and the eyelids with a particle filter (PF) and use several appearance metrics to calculate the likelihood of a given particle (candidate). Experimental results show the mean error to be greater than 3.5°.

Xie and Lin [78] infer the position of the eyeball center and other personal parameters using a simple one-target calibration. They calculate the gaze geometrically by using the iris center and eye corner positions on the image, with 2° accuracy in a single direction.

Chen and Ji [33] use a generic face model that includes several facial anchor points (nostrils, inner and outer eye corners) and one of the eyeball centers. After calibrating for the personal parameters, they track the facial points and fit the 3D model to estimate the gaze with 2.7° accuracy.

Yamazoe et al. [19, 79] segment the eye image pixels into three classes: skin, sclera, and iris. Using the segmentation results, they calculate the most possible eye pose by minimizing the projection errors for a given candidate. The accuracy of the system is reported to be around 9°.

Reale et al. [41] use the detected iris contours to calculate the eyeball center, and after calibrating for the visual axis-optical axis shift and the eyeball radius, they estimate the gaze direction. Finally, the most recent work in this category is from Heyman et al. [80], who employ canonical correlation analysis (CCA) to estimate the head pose in a similar manner to AAMs. They calibrate the eyeball radius during initialization and estimate the iris center using a segmentation method. Their system estimates the gaze direction with 5.64° accuracy.

4. Calibration Strategies

Traditionally, calibration of the eye trackers consists of asking the subject to look at several targets in known positions. In this way, either the personal parameters (e.g., angle between visual and optical axis of the eye, eyeball radius) or the camera parameters (e.g., focal length, position with respect to the display) are learned.

Several papers that we analyze in this work present novel techniques to make this process easier for the subject using the tracker. Yamazoe et al. [19, 79] employ a transparent calibration process, where the user does not need to be aware at all. They track the face over time to construct the 3D model of the face and eyes and start calculating the gaze when the calibration is ready. Alnajar et al. [50] use other users' gaze patterns to help estimate the current user's patterns. Sugano et al. [10] completely remove the need for training data and estimate the gaze in a probabilistic manner using computed saliency maps.

Another approach to collecting the training data without needing special actions from the user is to let the user operate the computer normally and take samples during mouse clicks [13, 46, 47]. This method is based on the assumption that the user looks around the mouse pointer while clicking.

Head movements constitute a challenge for eye tracker calibration, and even small movements may cause large errors in the estimations of a calibrated tracker. This holds true especially for appearance-based gaze trackers. Valenti et al.

Table 2: Publicly available datasets for remote, natural light gaze tracking.

	Year	# subjects	# targets	# head poses	Calibration	Resolution	Dataset size	References
UUlm	2007	20	2–9	19	Yes	1600 × 1200	2,200 imgs.	[103, 104]
HPEG	2009	10	Continuous	2	Yes	640 × 480	20 videos (~6.6 k imgs.)	[102, 105]
Gi4E	2012	103	12	1	No	800 × 600	1,236 imgs.	[106–108]
CAVE	2013	56	21	5	Yes	5184 × 3456	5,880 imgs.	[81, 100]
CVC	2013	12	12–15	4	Yes	1280 × 720	48 videos (~20 k imgs.)	[9, 98]
EYEDIAP	2014	16	Continuous	Continuous	Yes	1920 × 1080	94 videos	[109, 110]
Multiview	2014	50	160	8 (+synthesized)	Yes	1280 × 1024	64,000 imgs. (+synth.)	[51, 99]
MPIIGaze	2015	15	Continuous	Continuous	No	1280 × 720	213,659 imgs.	[13, 68]
OMEG	2015	50	10	Continuous	No	1280 × 1024	44,827 imgs.	[111]
TabletGaze	2015	51	35	Continuous	No	1280 × 720	816 videos (~120 k imgs.)	[67, 101]

[56] solve this problem by *retargeting* the calibration targets' positions to user's new field of view and calibrating the system again. Lu et al. [7, 29] require the user to record 5-second video clips while moving her/his head and use these to correct errors caused by head movements. Xie and Lin [78] require just a single target calibration, where the user keeps looking at the same position on the screen and moves her/his head around. Zhang et al. [13] take an approach based on large datasets and use other people's training data to calibrate a more accurate tracker.

Making the calibration process transparent for the user and collecting the required large amount of data are two conflicting objectives. In order to use the available training data to full extent, Williams et al. and Liang et al. [11, 14] use partially labeled data and annotate some of the unlabeled samples automatically. Ono et al. [45] create new samples by adding shifts while cropping the eye images, and in this way they can model the resulting appearance change and compensate for it while searching local samples. Lu et al. [48, 49] create synthetic training data by modeling the pixel flow around the eyes, whereas Sugano et al. [51] use 8 cameras to model a large part of the face in 3D and to generate training samples from previously unobserved head poses.

5. Dealing with Head Pose

Model-based visible light gaze tracking methods are normally invariant to head movements, assuming the preprocessing steps such as eye localization or model fitting do not fail. However, the same does not hold for the appearance-based and feature-based systems. As Lu et al. [29] demonstrate, the head movement not only adds a shift to the gaze angle, but also makes the calibration invalid by distorting the eye appearance for appearance-based methods.

The naive approach to solving the problem of head movements is adding more training data. Nguyen et al. [37, 53] propose repeating the calibration up to 10 times, while Lai et al. [8] require 34,000 training samples per user.

Zhang et al. [13] use a large dataset of previously collected images to train a feature-based gaze tracker. Here, training data collected from many subjects can be used in estimating the gaze for another person. Head pose invariance is achieved by incorporating the head pose angles into the feature set.

In other approaches [46, 47], the multipose training data is grouped according to head pose, and only a subset corresponding to the most similar head pose is used in the active calibration. To reduce the need for additional training data, Lu et al. [48] synthetically generate training samples for unseen head poses.

Instead of pouring more data into the system, another option is to apply compensations or small fixes to keep the current calibration working. Lu et al. [63] propose an eye image alignment scheme to undo the deformation in these images. In their other works [7, 29], they train regression for this task and combine it with a compensation for head rotation.

Valenti et al. [56] keep the calibration targets in a flexible representation and *retarget* these to the display coordinates whenever the head pose is changed and recalibrate their system.

Cheung and Peng [18] assume the PC-EC feature is completely invariant to head pose and apply only head rotation compensation in their system.

6. Available Datasets

Several papers that we analyzed contain a summary of publicly available datasets for visible light gaze tracking [13, 51, 81]. However, they are mostly for the purpose of comparison with the presented datasets in the mentioned work and thus may lack some pieces of related information.

In Table 2, we bring together all the datasets mentioned in these works (with several more recently published additions), in an attempt to provide a reference for future research in the field.

One of the datasets [82] cited in the previous reviews has been removed, as it provided data for a head mounted setup.

7. Gaze Tracking as a Service

While visible light gaze tracking has become a hot topic in the academia in recent years (as can be observed in Figure 2), the industry is not trailing far behind either. Here, we talk about several companies already providing gaze tracking service based on regular cameras found on consumer devices.

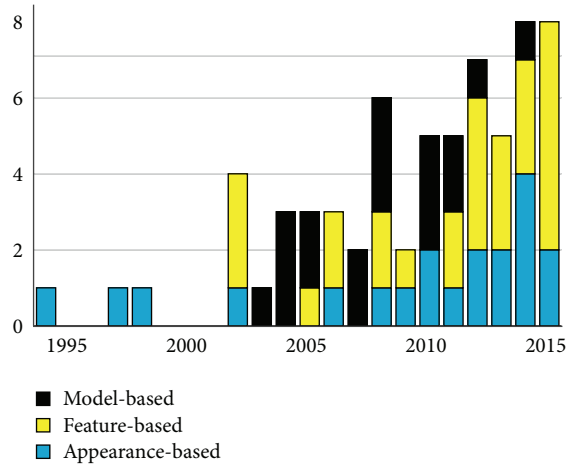


Figure 2: Number of works from different categories of eye trackers according to the publication year.

GazeHawk [83] (now closed) was enabling its customers to convey remote eye tracking studies inside the user's browser. xLabs [84] is another similar service, which is also available as a Chrome extension. With the extension, several demos (including continuous calibration by an ant smashing game) can be tried. Lastly, Sticky [85] also provides a JavaScript-based service, suggesting use cases such as online ad placement and web page optimization. As the only service with detailed specifications, their eye tracker provides an average accuracy of 2.4°.

Other possible clients for this type of eye tracker are the game or application developers. SentiGaze [86] provides an SDK for developers targeting the Windows platform. FaceTrack from Visage Technologies [87] provides a similar C++ SDK for developers, with augmented reality, view control in gaming, and view-dependent rendering suggested as possible use cases. The SDK provides detailed information such as mouth contour, chin pose, and eye openness, in addition to the gaze information. InSight SDK [88] takes one step further and combines the gaze information with mood, age, and gender estimation.

With the transition from desktop programs to mobile apps in recent years, two companies see a possibility for gaze tracking on this platform. Snapdragon [89] provides an SDK for Android apps, whereas Umoove [90] has a product on both iOS and Android platforms.

8. Open Source Projects

A few works that we analyze in this paper have released their source code with an open source license. In this section, we list these options so that new projects in the field will have a starting point for the codebase. Table 3 shows a summary of the listed projects.

Opengazer [91] is an eye tracker from Cambridge University, which is unfortunately no longer maintained. It uses Gaussian process regression with eye images as features, which is similar to the technique described by Nguyen et al.

Table 3: Open source gaze trackers and the related publications.

	Language	Platform	License	References
Opengazer	C/C++	Linux/Mac	GPLv2	[91]
NetGazer	C++/C#	Windows	GPLv2	[92]
CVC ET	C/C++	Linux/Mac	GPLv2	[9, 30, 93]
NNET	Objective C	iOS	GPLv3	[43, 44, 94]
EyeTab	Python/C++	Windows	MIT	[12, 95]
TurkerGaze	JavaScript	All	MIT	[31, 96]
Camgaze	Python	All	?	[97]

[37]. NetGazer [92] is the port of Opengazer for the Windows platform and is not maintained anymore either.

In the recent years, a fork of Opengazer project, named CVC Eye Tracker [93], was made available and is maintained actively by researchers from Universitat Autònoma de Barcelona. This project is the basis for two works analyzed in our review [9, 30].

Neural Network Eye Tracker (NNET) [94] is the NN-based eye tracker implementation for iPad devices, which is presented in two articles [43, 44]. EyeTab [95] is another open source codebase for tablet computers, which uses the iris contour model-based method described by Wood and Bulling [12].

Recently, the TurkerGaze project [31, 96] was made available on GitHub. This application is totally implemented in JavaScript (JS), which makes it platform independent (with possible extension to the mobile). The library has a polished interface for calibration and verification and comes with a small application for analyzing the gaze patterns recorded during conducted experiments. Although its proposed usage area is to enable crowdsourcing eye tracking tasks on platforms similar to Amazon Mechanical Turk, we believe it will have a larger impact on both academic works and web-based applications.

One last open source application is Camgaze [97], which is written in Python and calculates binocular gaze estimations.

9. Summary and Conclusions

In this work, we have tried to present a review of the state of the art in remote, natural light gaze trackers. Although in recent years many great works were published in the field, and the accuracy gap to reach the infrared-based trackers is closing, many open problems and unexplored approaches still remain.

Apart from the accuracy, the biggest challenges to these trackers are (a) making the calibration less painful for the user and (b) allowing free head movements. As we analyzed in the previous sections dedicated to these two problems, the field witnessed amazing works recently. Some open lines of work that we have identified in these areas are the following:

(i) *Maintaining Personal Calibration*. Most of the works we analyzed require some sort of calibration, be it for personal parameters for the user, for camera properties, or simply for training the gaze mapping

component. Although some techniques may already allow it (without stating explicitly), reusing the calibration information for the subsequent sessions of the same user is still pending extensive analysis. With such a technique, calibration before each session can be simplified or removed altogether.

(ii) *Using Calibration Data from Other Users*. Despite being explored in a few papers [13, 50], we believe the accumulation (or collection) of training data from people other than the current user will receive more focus in the coming years. This is analogous to training classifiers or detectors in other computer vision tasks, and it will let us make better use of the large datasets that we have begun to build.

(iii) *Other Ways of Collecting Data*. Collecting calibration samples each time the user clicks the mouse enabled us to create very large datasets for the first time [13, 46, 47]. Especially with the advent of JavaScript-based eye trackers [31], other possibilities such as remotely crowdsourcing data collection will emerge. Larger data will eventually let us explore previously impossible ideas, a trend which is common in computer vision.

These lines of work are mostly around the topic of data collection and calibration, and they will help solve the large data needs of training for different head poses.

Most of the recent high-performing techniques [11, 14, 28, 63] are using feature-based gaze estimation, which shows the promise of this category over appearance- or model-based methods. Figure 2 also shows this tendency, and the increase in feature-based methods can be observed clearly. Over the next years, we will probably see more examples of similar work with the following focus points:

(i) *Different Features*. The PC-EC vector, pixel intensity and color, and other standard features (such as HOG and LBP) have been used so far. New feature representations that may be better suited to the problem at hand will greatly improve the eye tracking accuracy. The desired characteristics of such features are (a) invariance to head pose, (b) invariance to intensity changes, and (c) invariance to personal appearance differences.

(ii) *Migrating Proven Ideas from Other CV Fields*. Use of convolutional neural networks (CNNs) [13], features such as HOG and LBP, and in general the computer vision (CV) pipeline [36] are changing our approach to the gaze tracking problem. These ideas were already commonplace in other areas of CV, and we believe our community will keep transferring insights which have been proven to work for other problems.

Apart from these technical challenges and lines of work, as a society, our biggest problems are related to transparency and letting others build on our work.

Firstly, only very few of these works report their accuracy on publicly available datasets or publish the dataset they use. This is a must in other computer vision areas so that the results from techniques can be compared and verified. Moreover, standardization of the processing pipeline will immediately follow (as it depends on the training data structure) and will foster our progress.

Our second problem is that only few works make their source code available. This prevents other researchers from *standing on the shoulders of giants* and hinders the rate of our progress. We believe that, by releasing our source code, we can create stronger ties and cooperation in the field.

In conclusion, the amount and quality of the recent work in the field are promising and signal even faster progress in the coming years. With this *map* of the current state of the art that you are holding in your hands (or gazing at through an electronic display), we hope to provide a reference point for all these amazing works we cannot wait to see.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Spanish Gov. Grants MICINN TIN2009-10435 and Consolider 2010 MIPRCV, Univ. Autònoma de Barcelona grants, and the Google Faculty Award.

References

[1] Tobii eye-trackers, October 2015, http://www.tobii.com/.

[2] SensoMotoric Instruments GmbH, October 2015, http://www.smivision.com/.

[3] The Eye Tribe, October 2015, http://theeyetribe.com/.

[4] Gaze Tracking Library, October 2015, http://sourceforge.net/projects/gazetrackinglib/.

[5] C. H. Morimoto and M. R. M. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[6] D. W. Hansen and Q. Ji, "In the eye of the beholder: a survey of models for eyes and gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478–500, 2010.

[7] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, "A head pose-free approach for appearance-based gaze estimation," in *Proceedings of the 22nd British Machine Vision Conference (BMVC '11)*, pp. 126.1–126.11, September 2011.

[8] C.-C. Lai, Y.-T. Chen, K.-W. Chen, S.-C. Chen, S.-W. Shih, and Y.-P. Hung, "Appearance-based gaze tracking with free head movement," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR '14)*, pp. 1869–1873, Stockholm, Sweden, August 2014.

[9] O. Ferhat, F. Vilariño, and F. J. Sánchez, "A cheap portable eye-tracker solution for common setups," *Journal of Eye Movement Research*, vol. 7, no. 3, article 2, 2014.

[10] 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.

[11] K. Liang, Y. Chahir, M. Molina, C. Tijus, and F. Jouen, "Appearance-based gaze tracking with spectral clustering and

semi-supervised Gaussian process regression," in *Proceedings of the Conference on Eye Tracking South Africa (ETSA '13)*, vol. 1, pp. 17–23, ACM, Cape Town, South Africa, August 2013.

[12] E. Wood and A. Bulling, "EyeTab: model-based gaze estimation on unmodified tablet computers," in *Proceedings of the 8th Symposium on Eye Tracking Research and Applications (ETRA '14)*, P. Qvarfordt and D. W. Hansen, Eds., pp. 207–210, ACM, Safety Harbor, Fla, USA, March 2014.

[13] 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 (CVPR '15)*, pp. 4511–4520, Boston, Mass, USA, June 2015.

[14] O. Williams, A. Blake, and R. Cipolla, "Sparse and semi-supervised visual mapping with the S3GP," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 230–237, June 2006.

[15] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[16] R. Stiefelhagen, J. Yang, and A. Waibel, *Tracking Eyes and Monitoring Eye Gaze*, 1997.

[17] E. Skodras, V. G. Kanas, and N. D. Fakotakis, "On visual gaze tracking based on a single low cost camera," *Signal Processing: Image Communication*, vol. 36, pp. 29–42, 2015.

[18] Y.-M. Cheung and Q. Peng, "Eye gaze tracking with a web camera in a desktop environment," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 4, pp. 419–430, 2015.

[19] H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe, "Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions," in *Proceedings of the Eye Tracking Research and Applications Symposium (ETRA '08)*, pp. 245–250, Santa Barbara, Calif, USA, March 2008.

[20] D. W. Hansen, J. P. Hansen, M. A. Nielsen, A. S. Johansen, and M. B. Stegmann, "Eye typing using Markov and active appearance models," in *Proceedings of the 6th IEEE Workshop on Applications of Computer Vision (WACV '02)*, pp. 132–136, Orlando, FL, USA, 2002.

[21] D. Bäck, *Neural network gaze tracking using web camera [Ph.D. dissertation]*, Linköping University, Linköping, Sweden, 2005.

[22] J. Zhu and J. Yang, "Subpixel eye gaze tracking," in *Proceedings of the 5th IEEE International Conference on Automatic Face Gesture Recognition*, pp. 131–136, IEEE, Washington, DC, USA, May 2002.

[23] D. Torricelli, M. Goffredo, S. Conforto, M. Schmid, and T. D'Alessio, "A novel neural eye gaze tracker," in *Proceedings of the 2nd International Workshop on Biosignal Processing and Classification—Biosignals and Sensing for Human Computer Interface (BPC '06)*, pp. 86–95, 2006.

[24] R. Valenti, N. Sebe, and T. Gevers, "Simple and efficient visual gaze estimation," in *Workshop on Multimodal Interactions Analysis of Users in a Controlled Environment (MIAUCE)*, ICMI, no. 3, 2008.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 886–893, IEEE, San Diego, Calif, USA, June 2005.

[26] F. Martinez, A. Carbone, and E. Pissaloux, "Gaze estimation using local features and non-linear regression," in *Proceedings of the 19th IEEE International Conference on Image Processing (ICIP '12)*, vol. 1, pp. 1961–1964, IEEE, Orlando, Fla, USA, October 2012.

[27] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[28] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Inferring human gaze from appearance via adaptive linear regression," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, Eds., pp. 153–160, IEEE, Barcelona, Spain, November 2011.

[29] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, "Learning gaze biases with head motion for head pose-free gaze estimation," *Image and Vision Computing*, vol. 32, no. 3, pp. 169–179, 2014.

[30] O. Ferhat, A. Llanza, and F. Vilariño, "A feature-based gaze estimation algorithm for natural light scenarios," in *Pattern Recognition and Image Analysis*, R. Paredes, J. S. Cardoso, and X. M. Pardo, Eds., vol. 9117 of *Lecture Notes in Computer Science*, pp. 569–576, 2015.

[31] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "TurkerGaze: crowdsourcing saliency with webcam based eye tracking," http://arxiv.org/abs/1504.06755.

[32] J.-G. Wang, E. Sung, and R. Venkateswarlu, "Eye gaze estimation from a single image of one eye," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, pp. 136–143, IEEE, October 2003.

[33] J. Chen and Q. Ji, "3D gaze estimation with a single camera without IR illumination," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, IEEE, December 2008.

[34] T. Ishikawa, S. Baker, I. Matthews, and T. Kanade, "Passive driver gaze tracking with active appearance models," in *Proceedings of the 11th World Congress on Intelligent Transportation Systems (ITS '04)*, vol. 3, Nagoya, Japan, October 2004.

[35] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artificial neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '94)*, pp. 1–14, January 1994.

[36] Y. Zhang, A. Bulling, and H. Gellersen, "Towards pervasive eye tracking using low-level image features," in *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*, C. H. Morimoto, H. O. Istance, S. N. Spencer, J. B. Mulligan, and P. Qvarfordt, Eds., pp. 261–264, ACM, Santa Barbara, Calif, USA, March 2012.

[37] B. L. Nguyen, Y. Chahir, M. Molina, C. Tijus, and F. Jouen, "Eye gaze tracking with free head movements using a single camera," in *Proceedings of the Symposium on Information and Communication Technology (SoICT '10)*, N. T. Giang, N. T. Hai, and H. Q. Thang, Eds., vol. 449 of *ACM International Conference Proceeding Series*, pp. 108–113, ACM, Hanoi, Vietnam, August 2010.

[38] K.-H. Tan, D. J. Kriegman, and N. Ahuja, "Appearance-based eye gaze estimation," in *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision (WACV '02)*, pp. 191–195, IEEE Computer Society, 2002.

[39] L. Sesma, A. Villanueva, and R. Cabeza, "Evaluation of pupil center-eye corner vector for gaze estimation using a web cam," in *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*, C. H. Morimoto, H. O. Istance, S. N. Spencer, J. B. Mulligan, and P. Qvarfordt, Eds., pp. 217–220, ACM, Santa Barbara, Calif, USA, March 2012.

[40] H. Wu, Y. Kitagawa, T. Wada, T. Kato, and Q. Chen, "Tracking iris contour with a 3D eye model for gaze estimation," in *Computer Vision—ACCV 2007: 8th Asian Conference on Computer Vision, Tokyo, Japan, November 18–22, 2007, Proceedings, Part*

I, Y. Yagi, S. B. Kang, I.-S. Kweon, and H. Zha, Eds., vol. 4843 of *Lecture Notes in Computer Science*, pp. 688–697, Springer, Berlin, Germany, 2007.

[41] M. Reale, T. Hung, and L. Yin, "Viewing direction estimation based on 3D eyeball construction for HRI," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition—Workshops (CVPRW '10)*, pp. 24–31, IEEE, San Francisco, Calif, USA, June 2010.

[42] L.-Q. Xu, D. Machin, and P. Sheppard, "A novel approach to real-time non-intrusive gaze finding," in *Proceedings of the British Machine Vision Conference*, J. N. Carter and M. S. Nixon, Eds., British Machine Vision Association, Southampton, UK, 1998.

[43] W. Sewell and O. Komogortsev, "Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network," in *Proceedings of the Extended Abstracts on Human Factors in Computing Systems (CHI EA '10)*, E. D. Mynatt, D. Schoner, G. Fitzpatrick, S. E. Hudson, W. K. Edwards, and T. Rodden, Eds., pp. 3739–3744, ACM, Atlanta, Ga, USA, April 2010.

[44] C. Holland and O. V. Komogortsev, "Eye tracking on unmodified common tablets: challenges and solutions," in *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*, C. H. Morimoto, H. O. Istance, S. N. Spencer, J. B. Mulligan, and P. Qvarfordt, Eds., pp. 277–280, ACM, Santa Barbara, Calif, USA, March 2012.

[45] Y. Ono, T. Okabe, and Y. Sato, "Gaze estimation from low resolution images," in *Advances in Image and Video Technology*, L.-W. Chang and W.-N. Lie, Eds., vol. 4319 of *Lecture Notes in Computer Science*, pp. 178–188, Springer, 2006.

[46] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "An incremental learning method for unconstrained gaze estimation," in *Computer Vision—ECCV 2008*, vol. 5304 of *Lecture Notes in Computer Science*, pp. 656–667, Springer, Berlin, Germany, 2008.

[47] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "Appearance-based gaze estimation with online calibration from mouse operations," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 6, pp. 750–760, 2015.

[48] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Head pose-free appearance-based gaze sensing via eye image synthesis," in *Proceedings of the 21st IEEE International Conference on Pattern Recognition (ICPR '12)*, pp. 1008–1011, Tsukuba, Japan, November 2012.

[49] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Gaze estimation from eye appearance: a head pose-free method via eye image synthesis," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3680–3693, 2015.

[50] F. Alnajar, T. Gevers, R. Valenti, and S. Ghebreab, "Calibration-free gaze estimation using human gaze patterns," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 137–144, IEEE, Sydney, Australia, December 2013.

[51] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3D gaze estimation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1821–1828, IEEE, Columbus, Ohio, USA, June 2014.

[52] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Mass, USA, 2006.

[53] B. L. Nguyen, "Eye gaze tracking," in *Proceedings of the IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF '09)*, pp. 1–4, IEEE, Da Nang, Vietnam, July 2009.

[54] D. W. Hansen, M. Nielsen, J. P. Hansen, A. S. Johansen, and M. B. Stegmann, "Tracking eyes using shape and appearance," in *Proceedings of the IAPR Workshop on Machine Vision Applications (MVA '02)*, pp. 201–204, December 2002.

[55] R. Valenti, J. Staiano, N. Sebe, and T. Gevers, "Webcam-based visual gaze estimation," in *Image Analysis and Processing—ICIAP 2009*, P. Foggia, C. Sansone, and M. Vento, Eds., vol. 5716 of *Lecture Notes in Computer Science*, pp. 662–671, Springer, 2009.

[56] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 802–815, 2012.

[57] S.-J. Baek, K.-A. Choi, C. Ma, Y.-H. Kim, and S.-J. Ko, "Eyeball model-based iris center localization for visible image-based eye-gaze tracking systems," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 2, pp. 415–421, 2013.

[58] D. Torricelli, S. Conforto, M. Schmid, and T. D'Alessio, "A neural-based remote eye gaze tracker under natural head motion," *Computer Methods and Programs in Biomedicine*, vol. 92, no. 1, pp. 66–78, 2008.

[59] I. F. Ince and J. W. Kim, "A 2D eye gaze estimation system with low-resolution webcam images," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, article 40, 2011.

[60] P. Nguyen, J. Fleureau, C. Chamaret, and P. Guillotel, "Calibration-free gaze tracking using particle filter," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '13)*, pp. 1–6, IEEE, San Jose, Calif, USA, July 2013.

[61] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2106–2113, Kyoto, Japan, October 2009.

[62] A. Wojciechowski and K. Fornalczyk, "Exponentially smoothed interactive gaze tracking method," in *Computer Vision and Graphics: International Conference, ICCVG 2014, Warsaw, Poland, September 15–17, 2014. Proceedings*, L. J. Chmielewski, R. Kozera, B.-S. Shin, and K. W. Wojciechowski, Eds., vol. 8671 of *Lecture Notes in Computer Science*, pp. 645–652, Springer, Berlin, Germany, 2014.

[63] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2033–2046, 2014.

[64] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[65] M. Heikkil, M. Pietikinen, and C. Schmid, "Description of interest regions with center-symmetric local binary patterns," in *Computer Vision, Graphics and Image Processing: 5th Indian Conference, ICVGIP 2006, Madurai, India, December 13–16, 2006. Proceedings*, P. K. Kalra and S. Peleg, Eds., vol. 4338 of *Lecture Notes in Computer Science*, pp. 58–69, Springer, Berlin, Germany, 2006.

[66] T. Schneider, B. Schauerte, and R. Stiefelhagen, "Manifold alignment for person independent appearance-based gaze estimation," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR '14)*, pp. 1167–1172, IEEE, Stockholm, Sweden, August 2014.

[67] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "TabletGaze: unconstrained appearance-based gaze estimation in mobile tablets," http://arxiv.org/abs/1508.01244.

[68] MPIIGaze Dataset, October 2015, https://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/gaze-based-human-computer-interaction/appearance-based-gaze-estimation-in-the-wild/.

[69] A. W. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least squares fitting of ellipses," in *Proceedings of the 13th International Conference on Pattern Recognition (ICPR '96)*, pp. 253–257, IEEE, Vienna, Austria, August 1996.

[70] J.-G. Wang, E. Sung, and R. Venkateswarlu, "Estimating the eye gaze from one eye," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 83–103, 2005.

[71] D. W. Hansen and A. E. C. Pece, "Eye typing off the shelf," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 159–164, June-July 2004.

[72] D. W. Hansen and A. E. C. Pece, "Eye tracking in the wild," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 155–181, 2005.

[73] H. Wu, Q. Chen, and T. Wada, "Conic-based algorithm for visual line estimation from one image," in *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 260–265, IEEE, May 2004.

[74] S.-C. Huang, Y.-L. Wu, W.-C. Hung, and C.-Y. Tang, "Point-of-regard measurement via iris contour with one eye from single image," in *Proceedings of the IEEE International Symposium on Multimedia (ISM '10)*, pp. 336–341, IEEE, Taichung, Taiwan, December 2010.

[75] W. Zhang, T.-N. Zhang, and S.-J. Chang, "Gazing estimation and correction from elliptical features of one iris," in *Proceedings of the 3rd International Congress on Image and Signal Processing (CISP '10)*, pp. 1647–1652, Yantai, China, October 2010.

[76] T. Fukuda, K. Morimoto, and H. Yamana, "Model-based eye-tracking method for low-resolution eye-images," in *Proceedings of the 2nd Workshop on Eye Gaze in Intelligent Human Machine Interaction*, Palo Alto, Calif, USA, 2011.

[77] M. R. Mohammadi and A. Raie, "Robust pose-invariant eye gaze estimation using geometrical features of iris and pupil images," in *Proceedings of the 20th Iranian Conference on Electrical Engineering (ICEE '12)*, pp. 593–598, Tehran, Iran, May 2012.

[78] J. Xie and X. Lin, "Gaze direction estimation based on natural head movements," in *Proceedings of the Fourth International Conference on Image and Graphics (ICIG '07)*, pp. 672–677, Sichuan, China, August 2007.

[79] H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe, "Remote and head-motion-free gaze tracking for real environments with automated head-eye model calibrations," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '08)*, pp. 1–6, Anchorage, Alaska, USA, June 2008.

[80] T. Heyman, V. Spruyt, and A. Ledda, "3D Face tracking and gaze estimation using a monocular camera," in *Proceedings of the 2nd International Conference on Positioning and Context-Awareness (PoCA '11)*, pp. 23–28, Brussels, Belgium, 2011.

[81] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: passive eye contact detection for human-object interaction," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*, pp. 271–280, ACM, 2013.

[82] C. D. McMurrough, V. Metsis, D. Kosmopoulos, I. Maglogiannis, and F. Makedon, "A dataset for point of gaze detection using head poses and eye images," *Journal on Multimodal User Interfaces*, vol. 7, no. 3, pp. 207–215, 2013.

[83] GazeHawk, October 2015, http://www.gazehawk.com/.

[84] xLabs eye, gaze and head tracking, October 2015, http://xlabsgaze.com/.

[85] Sticky, 2015, http://www.sticky.ad/technical-details.html.

[86] SentiGaze SDK, October 2015, http://www.neurotechnology.com/sentigaze.html.

[87] FaceTrack—Visage Technologies, 2015, https://visagetechnologies.com/products-and-services/visagesdk/facetrack/.

[88] InSight SDK, October 2015, http://sightcorp.com/insight/.

[89] Snapdragon SDK for Android, October 2015, https://developer.qualcomm.com/software/snapdragon-sdk-android.

[90] Umoove, 2015, http://www.umoove.me/.

[91] Opengazer: open-source gaze tracker for ordinary webcams (software), October 2015, http://www.inference.phy.cam.ac.uk/opengazer/.

[92] NetGazer, 2015, http://sourceforge.net/projects/netgazer/.

[93] CVC Eye Tracker, October 2015, https://github.com/tiendan/OpenGazer.

[94] Neural Network Eye Tracker (NNET), October 2015, http://cs.txstate.edu/~ok11/nnet.html.

[95] EyeTab, October 2015, https://github.com/errollw/EyeTab/.

[96] TurkerGaze, October 2015, https://github.com/PrincetonVision/TurkerGaze.

[97] Camgaze, 2015, https://github.com/wallarelvo/camgaze.

[98] CVC Eye Tracking DB, October 2015, http://mv.cvc.uab.es/projects/eye-tracker/cvceyetrackerdb.

[99] Multi-View Gaze Dataset, October 2015, http://www.hci.iis.u-tokyo.ac.jp/datasets/.

[100] Columbia Gaze Data Set, October 2015, http://www.cs.columbia.edu/CAVE/databases/columbia_gaze/.

[101] Rice TabletGaze Dataset, October 2015, http://sh.rice.edu/tablet_gaze.html.

[102] Head Pose and Eye Gaze (HPEG) Dataset, October 2015, http://sspnet.eu/2010/02/head-pose-and-eye-gaze-hpeg-dataset/.

[103] UUlm Head Pose and Gaze Database, 2015, https://www.uni-ulm.de/in/neuroinformatik/mitarbeiter/g-layher/image-databases.html.

[104] U. Weidenbacher, G. Layher, P.-M. Strauss, and H. Neumann, "A comprehensive head pose and gaze database," in *Proceedings of the 3rd IET International Conference on Intelligent Environments (IE '07)*, pp. 455–458, September 2007.

[105] S. Asteriadis, D. Soufleros, K. Karpouzis, and S. Kollias, "A natural head pose and eye gaze dataset," in *Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots (AFFINE '09)*, pp. 1:1–1:4, ACM, Cambridge, Mass, USA, November 2009.

[106] Gi4E Database, October 2015, http://gi4e.unavarra.es/databases/gi4e/.

[107] V. Ponz, A. Villanueva, and R. Cabeza, "Dataset for the evaluation of eye detector for gaze estimation," in *Proceedings of the ACM Conference on Ubiquitous Computing (UbiComp '12)*, pp. 681–684, Pittsburgh, Pa, USA, September 2012.

[108] A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza, "Hybrid method based on topography for robust detection of iris center and eye corners," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 9, no. 4, article 25, 2013.

[109] EYEDIAP Dataset, October 2015, http://www.idiap.ch/dataset/eyediap.

[110] K. A. Funes 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 8th Symposium on Eye Tracking Research and Applications (ETRA '14)*, pp. 255–258, ACM, March 2014.

[111] Q. He, X. Hong, X. Chai et al., "OMEG: Oulu multi-pose eye gaze dataset," in *Proceedings of the 19th Scandinavian Conference on Image Analysis (SCIA '15), Copenhagen, Denmark, June 2015*, R. R. Paulsen and K. S. Pedersen, Eds., vol. 9127 of *Lecture Notes in Computer Science*, pp. 418–427, Springer, 2015.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 7496735, 10 pages
http://dx.doi.org/10.1155/2016/7496735

Research Article

Learning-Based Visual Saliency Model for Detecting Diabetic Macular Edema in Retinal Image

Xiaochun Zou,[1] Xinbo Zhao,[2] Yongjia Yang,[2] and Na Li[2]

[1]*School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China*
[2]*School of Computer Science, Northwestern Polytechnical University, Chang'an Campus, P.O. Box 886, Xi'an, Shaanxi 710129, China*

Correspondence should be addressed to Xinbo Zhao; xbozhao@nwpu.edu.cn

Received 9 October 2015; Accepted 16 December 2015

Academic Editor: Francesco Camastra

This paper brings forth a learning-based visual saliency model method for detecting diagnostic diabetic macular edema (DME) regions of interest (RoIs) in retinal image. The method introduces the cognitive process of visual selection of relevant regions that arises during an ophthalmologist's image examination. To record the process, we collected eye-tracking data of 10 ophthalmologists on 100 images and used this database as training and testing examples. Based on analysis, two properties (Feature Property and Position Property) can be derived and combined by a simple intersection operation to obtain a saliency map. The Feature Property is implemented by support vector machine (SVM) technique using the diagnosis as supervisor; Position Property is implemented by statistical analysis of training samples. This technique is able to learn the preferences of ophthalmologist visual behavior while simultaneously considering feature uniqueness. The method was evaluated using three popular saliency model evaluation scores (AUC, EMD, and SS) and three quality measurements (classical sensitivity, specificity, and Youden's J statistic). The proposed method outperforms 8 state-of-the-art saliency models and 3 salient region detection approaches devised for natural images. Furthermore, our model successfully detects the DME RoIs in retinal image without sophisticated image processing such as region segmentation.

1. Introduction

Diabetes is a chronic disease that can cause many serious complications including diabetic retinopathy (DR, damage to the retina). DR is an important cause of blindness. One percent of global blindness can be attributed to DR [1]. Diabetic macular edema (DME) is the most common cause of visual loss in DR, which is due to leaking of fluid from blood vessels within the macula. Fortunately, the Early Treatment Diabetic Retinopathy Study (ETDRS) has been able to reduce moderate vision loss in patients with clinically significant macular edema (CSME) by approximately 50% [2]. Hence to prevent vision loss, early diagnosis of DME is very important. Because there are no visual symptoms in the early stages of DME, the retinal fundus images are recommended in the diagnosis and treatment. The fundus image analysis helps the ophthalmologists in understanding the onset and assessment of the diseases. A reliable determination of clinically meaningful regions of interest (RoIs) in retinal image is at the very base of strategies for DME diagnosis. The advent of new inexpensive fundus cameras and rapid growth in information technology has made the automated system for DME RoIs selection possible. Such a tool is going to be notably useful in health camps particularly, especially in rural areas in developing countries wherever an outsized population laid low with these diseases goes unknown.

Exudates are the single most important retinal lesion detectable in retinal images. However, hard and yellow exudates within 500 μm of the center of the fovea with adjacent retinal thickening indicate the presence of clinically significant macular edema (CSME), as defined by ETDRS [2]. But automatic DME RoIs finding in retinal images is a very challenging task. Because other retinal features such as blood vessels and outside diameter (OD) of bulbus oculi also have the similar brightness patterns and gray level variations, the naive use of current low-level-RoI-extraction methods for retinal images would probably fail.

Nevertheless, the ophthalmologists are always capable of figuring out a very precise diagnosis. When they search for CSME in retinal image, attention helps them rapidly disregard the "usual" and find the "unusual" visual elements. Some computational models of attention have been proposed to predict where people look in natural scenes [3–5]. Though the existing saliency models do well qualitatively, the models have limited use in DME RoIs detection because they frequently do not match actual ophthalmologists' precise diagnosis (ground truth, GT).

In this paper, we propose three contributions to DME detecting. First, we introduce the computational visual saliency models in retinal images in the context of DME detection. Through this model, we emulate the ophthalmologists' first examination step where she/he defines and separates high informative DME diagnostic regions. Second, by analyzing the precise diagnosis, we choose only a few concepts that encompass comprehensive ophthalmologists' visual behaviors, clarify the interactions among them, and develop a method for implementing the visual saliency. The method combines the advantages of a low-level image characterization with a high discriminant power in terms of DME tissular and spatial properties, information learned from the ophthalmologists. Third, we show that our model which is able to detect the DME RoIs in retinal images outperforms the mainstream salient region detection schemes.

This paper is organized as follows. Section 2 provides a brief description and discussion of some previous works. Section 3 is devoted to description of the implementation of the model. In Section 3.1, we present the images, eye-tracking data, and ground truth data for saliency model research. Section 3.2 describes the properties, derivations, and relationships of salience concepts. The detailed description of our model is in Section 3.3. Section 4 evaluates our approach using three popular saliency model evaluation scores (AUC, EMD, and SS) and three quality measurements (classical sensitivity, specificity, and Youden's J statistic) with 8 state-of-the-art saliency models and 3 salient region detection approaches. The conclusions and perspectives are discussed in Section 5.

2. Related Work

The problem of automatically detecting DME RoIs in retinal image has been approached using many techniques [6]. Phillips et al. [7] have proposed a method for the quantification of diabetic maculopathy using fluorescein angiograms. A combination of shade correction and thresholding techniques was used for preprocessing. The exudates were then detected by thresholding which was calculated based on the distribution of gray levels in the image. Hunter et al. [8] used feature extraction and classification techniques for the automated diagnosis of referable maculopathy. The technique detects and filters the candidate points with strong local contrast. Segmentation of candidate regions was carried out next in order to find the location of lesions. The lesions were distinguished from nonlesions by feature extraction technique. Authors have used shape, color, and texture of the candidate and the contrast between it and the surrounding retinal background. A multilayer perceptron (MLP) was used as classifier which classifies the lesions as dark or bright. A two-stage methodology was used for the detection and classification of DME severity of fundus images [9]. The first step was a supervised learning approach by which the fundus images were classified as normal or abnormal. By examining the symmetry of macular region using a rotational asymmetry metric the abnormal fundus images were further classified into moderate and severe DME. Osareh et al. [10] used an automatic method for the classifications of the regions into exudates and nonexudates patches using a neural network. The fundus images were preprocessed using color normalization and contrast enhancement techniques. The images were segmented next into homogenous regions using fuzzy C-means clustering. Based on the location of the exudates at the macular region Lim et al. [11] have classified the fundus images into normal, stage 1, and stage 2 of DME. The exudates were extracted from the fundus images using a marker controlled watershed transformation.

A DME pathologic diagnosis is the result of a complex series of activities mastered by the ophthalmologists. Classical psychophysical theories suggest that complex visual tasks, such as ophthalmologist examination, involve high degrees of visual attention [12].

Today, many saliency models based on a variety of techniques with compelling performance exist. One of the most influential ones is a pure bottom-up attention model proposed by Itti et al. [3], based on the feature integration theory [13]. In this theory, an image is decomposed into low-level attributes such as color, intensity, and orientation. Based on the idea of decorrelation of neural responses, Diaz et al. [14] proposed an effective model of saliency known as Adaptive Whitening Saliency (AWS). Another class of models is based on probabilistic formulation. Torralba [4] proposed a Bayesian framework for visual search which is also applicable for saliency detection. Similarly, Zhang et al. [5] proposed SUN (Saliency Using Natural statistics) model in which bottom-up saliency emerges naturally as the self-information of visual features. Graph Based Visual Saliency (GBVS) [15] is another method based on graphical models. Machine learning approaches have also been used in modeling visual attention by learning models from recorded eye-fixations. For learning saliency, Kienzle et al. [16] and Tilke et al. [17] used image patches and a vector of several features at each pixel, respectively.

These computational models have been used to characterize RoIs in natural images, but their use in medical images has remained very limited. Jampani et al. [18] investigate the relevance of computational saliency models in medical images in the context of abnormality detection. Saliency maps were computed using three popular models: ITTI [3], GBVS [15], and SR [19]. Gutiérrez et al. [20] have developed a visual model for finding regions of interest in basal cell carcinoma images that has three main components: segmentation, saliency detection, and competition. The key insight from these studies is that saliency continues to play a predominant role in examining medical images.

3. Learning a Saliency Model for DME RoIs Detection

3.1. Database of Eye-Tracking Data. For learning the preferences of ophthalmologist visual behavior and recording their eye-tracking data, we established an eye-tracking database, called EDMERI database (eye-tracking database for detecting diabetic macular edema in retinal image). The EDMERI allows quantitative analysis of fixation points and provides ground truth data for saliency model research. Compared with several eye-tracking datasets that are publicly available, the main motivation of our new dataset is for detecting diabetic retinopathy region in retinal images.

The purpose of the current analysis was to model the cognitive process of visual selection of relevant regions that arises during detecting diabetic macular edema in retinal image. This reinforces our assumption that DME is a visible CSME feature. Under this constraint, we collected 100 images with DME (e.g., Figure 1(a)) from DIARETDB0 [6], DIARETDBI1 [6], MESSIDOR [21], and STARE [22], which are four standard diabetic retinopathy databases. These images stored in JPEG format were resized to 1152×1500 resolution. And we recorded eye-tracking data from ten expert ophthalmologists, with at least six years of experience, who were asked to view these images to find diabetic retinopathy regions. We used a Tobii TX300 Eye Tracker device to record eye movements at a sample rate of unique combination of 300 Hz. It has very high precision and accuracy and robust eye tracking and compensation for large head movements extends the possibilities for unobtrusive research of oculomotor functions and human behavior. A variety of researcher profiles, including ophthalmologists, can use the system without needing extensive training.

In the experiments, each image was presented for 10 s and followed by a rapid and automatic calibration procedure. To ensure high-quality tracking results, we checked camera calibration every 10 images. During first 1 s viewing, ophthalmologists maybe free viewed the histopathological image, so we discarded the first 1 s viewing tracking results of each ophthalmologist. In order to obtain a continuous ground truth of an image from the eye-tracking data of a user, we convolved a Gaussian filter across the user's fixation locations, similar to the "landscape map." We overlapped the eye-tracking data collected from all ophthalmologists (e.g., Figure 1(b)) and then generated ground truth of the average locations (e.g., Figure 1(c)).

3.2. Relevant Properties and Bayesian Formulation. In this subsection, we will discuss the relevant properties of the visual saliency concepts in DME RoIs detection we have considered and the relationship among them. We assume that saliency values in a retinal image are relative to at least two properties, as described below.

Feature Property (FP) (Saliency Is Relative to the Strength of Features in the Pixel). In nature scene, features are traditionally separated into two types, high- and low-level features. High-level features include face, text, and events. Low-level features include intensity, color, regional contrast, and orientations. Since high-level features are more complex to define and extract in a retinal image, we only consider low-level features in this paper. For example, when the ophthalmologists are diagnosing the presence of DME, a pixel with strong yellow color feature tends to be more significant than one with weak ones.

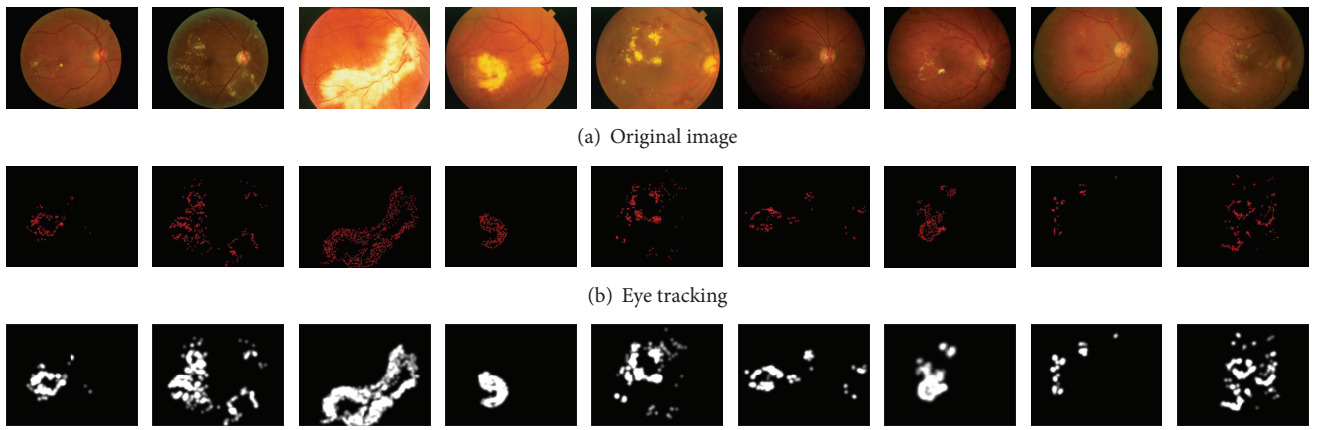
Position Property (PP) (Saliency Is Relative to the Location of the Pixel in the Image). Actual ophthalmologists' precise diagnoses have shown that DME always appears within 500 μm of the center of the fovea. That means the probability distribution of saliency for every pixel in a retinal image has a strong center bias property, so locational preferences in DME RoIs detection will be considered in our approach.

Considering the properties described above, the saliency of a pixel can be defined as the probability of saliency given the features and positions. Denote $F_X = [f_X^1, f_X^2, \ldots, f_X^n]$, $X = [x, y] \in I$, as a feature set including n features located in a pixel position X of image I. The saliency value can be denoted as $P(S \mid X, F_X)$, where "S equal 1" indicates that this pixel X is salient (i.e., it is in DME RoIs) and zero otherwise. Based on the assumption that features can appear in all spatial locations, we assume X and F_X are independent of each other, as Zhang et al. [5] did. The probability of pixel X being salient can be written as

$$\begin{aligned} P(S \mid X, F_X) &= \frac{P(X, F_X \mid S) \cdot P(S)}{P(X, F_X)} \\ &= \frac{P(X \mid S) \cdot P(F_X \mid S) \cdot P(S)}{P(X) \cdot P(F_X)} \\ &= \frac{P(S \mid X)}{P(S)} \cdot \frac{P(S \mid F_X)}{P(S)} \cdot P(S) \\ &= \frac{P(S \mid X) \cdot P(S \mid F_X)}{P(S)} \\ &\propto P(S \mid X) \cdot P(S \mid F_X). \end{aligned} \tag{1}$$

In (1), the term $P(X \mid S)$ is the probability of saliency given a position X and it corresponds to Position Property (PP). $P(S \mid F_X)$ is the probability of saliency of the features appearing in location X and it corresponds to Feature Property (FP). As a result, the probability of saliency is clearly relative to two terms: PP and FP.

3.3. Learning-Based Saliency Model. In contrast to manually designed measures of saliency, we follow a learning approach by using statistics and machine learning methods directly from eye-tracking data. Based on (1), when the ophthalmologists are diagnosing the presence of DME, there are two terms that affect saliency value in a pixel of a retinal image: FP and PP. Between these, FP can be learned from training samples using SVM; PP can be learned from ground truth of training images using statistical method. As shown in Figure 2, a set of low-level visual features are extracted from some training images. After the feature extraction process, the features of the top 20% (bottom 50%) points in the ground truth are selected as training samples in each training image. All of



(a) Original image

(b) Eye tracking

(c) Ground truth

Figure 1: We collected eye-tracking data on 100 retinal images with diabetic macular edema from 10 ophthalmologists. Gaze tracking paths and fixation locations are recorded in (b). A continuous ground truth (b) is found by convolving a Gaussian over the fixation locations of all users.

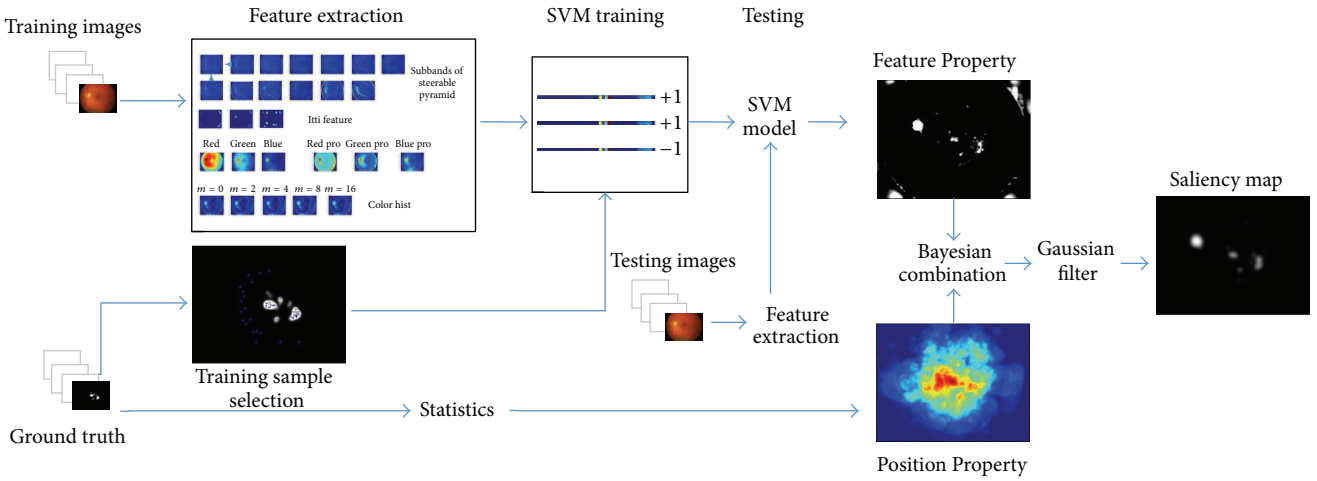


Figure 2: Illustration of learning process and saliency computing process. Feature Property: a set of low-level visual features are extracted from some training images. Feature vectors corresponding to the top 20% (bottom 50%) of the ophthalmologists' precise diagnoses (ground truth) are assigned +1(0) labels. Then a SVM classifier is trained from these features and is used for predicting DME on a test image. Position Property: we used a statistical analysis method to obtain it from ground truth. Finally, we combine Feature Property and Position Property adopting.

the training samples are sent to train a SVM model. Then, a test image can be decomposed into several feature maps and imported into SVM model to predict the FP. At the same time, PP also can be obtained from training images and their ground truth by statistical analysis. Finally, the two parts are combined, and a saliency map can be obtained after being convoluted with a Gaussian filter.

Features Extraction. After analyzing the DME dataset, we first extract a set of features for every pixel in each image including $m \times n$ pixels. Here, we use low-level features as they have already been shown to correlate with visual attention and have underlying biological plausibility [13, 23].

These features are listed below:

(i) Because they are physiologically plausible and have been shown to correlate with visual attention, we use the local energy of the steerable pyramid filters [24] as features. We currently find the pyramid subbands in four orientations and three scales, altogether thirteen images.

(ii) Traditionally, intensity, orientation, and color have been used as important features for saliency, derivation over static images. We include the three channels corresponding to these image features as calculated by Itti's saliency method [12].

(iii) We include three values of the red, green, and blue color channels as well as three features corresponding to probabilities of each of these color channels and five probabilities of each color as computed from 3D color histograms of the image filtered with a median filter at six different scales.

Eventually, all features are augmented in a 27D vector and are fed to classifiers explained in the next subsection. Each feature map is linearized into a $1 \times mn$ vector (similarly for class labels).

Feature Property. In (1), FP is the relationship between a given feature set F_X appearing in position X and saliency value S. One of the simplest methods to determine saliency is to average all the feature values. However, some features may be more important than others, so giving the same weight to all features is not appropriate and will give poor results. Instead, we use SVM to implement Feature Property.

We compile a large training set by sampling images at diagnosis. Each sample contains features at one point along with a +1/−1 label. Positive samples are taken from the top p percent salient pixels of the precise diagnosis and negative samples are taken from the bottom q percent. We chose samples from the top 20% and bottom 50% in order to have samples that were strongly positive and strongly negative. We avoided samples on the boundary between the two. We did not choose any samples within 10 pixels of the boundary of image. We trained models using ratios of negative to positive samples ranging from 1 to 5 and detected no change in the resulting ROC curves, so we chose to use a ratio of 1 : 1. Training feature vectors were normalized to have zero mean and unit standard deviation and the same parameters were used to normalize test data. To evaluate our model, we followed the 10-fold cross validation method. The method partitions the database into ten subsets randomly, each with M images. Every subset is selected sequentially as a test set and the remainders serve as the training set. Each time we trained the model from 9 parts and tested it over the remaining part. Results are then averaged over all partitions.

We used the LIBLINEAR support vector machine [25], a publicly available MATLAB version of SVM, to implement FP. We adopted linear kernels as they are faster and perform as well as nonlinear polynomial and RBF kernels for our specific task. In testing, instead of predicted labels (i.e., +1/−1), we use the value of $W^T f + b$, where W and b are learned parameters. We set the misclassification cost c at 1 and found that performance was the same for $c = 1$ to $c = 10{,}000$ and decreased when smaller than 1.

Position Property. As shown in (1), PP presents precise diagnosis preference for locations in an image. We implemented PP using a simple statistical method: sum up the values in the same position of density maps of database images, and normalize the result from zero to one. In the experiments, we denoted M_i as the ith ground truth image. Then the PP matrix M_p can be computed in

$$M_p = \frac{\sum_{i \in \text{dataset}} M_i - \min\left(\sum_{i \in \text{dataset}} M_i\right)}{\max\left(\sum_{i \in \text{dataset}} M_i\right) - \min\left(\sum_{i \in \text{dataset}} M_i\right)}. \quad (2)$$

Based on the finding, the center prior in which the majority of fixations happen near the center of the image can be observed in M_p.

Property Combination. The two matrices FP and PP, which are denoted as M_f and M_p, respectively, are combined in the saliency map M_s by an intersection operation and a convolution operation, as shown in

$$M_s = \left(M_f \bullet M_p\right) * \text{GF}. \quad (3)$$

Here $\bullet$ denotes a Hadamard product and $*$ denotes convolution operation. We set the parameter of the Gaussian filter δ at 10 in the EDMERI database.

4. Experiment and Result

We validate our model by applying it to two problems: (1) DME RoIs prediction and (2) segmentation of the DME RoIs in a retinal image. We used EDMERI databases to evaluate our results; the size of each image was 1152×1500 pixels. We chose 100 training samples from each of the training images, for a total of 10,000 training samples. The database provided ophthalmologists' eye-tracking data as ground truth.

Since there is no consensus over a unique score for saliency model evaluation, we report results over three, including Area Under the ROC Curve (AUC), Earth Movers Distance (EMD), and Similarity Score (SS). A model that performs well should have good overall scores.

AUC. It is the most widely used metric for evaluating visual saliency. Using this score, the model's saliency map is treated as a binary classifier on every pixel in the image; pixels with larger saliency values than a threshold are classified as fixated while the rest of the pixels are classified as nonfixated [26]. Precise diagnoses are used as ground truth. By varying the threshold, the ROC curve is drawn as the false positive rate versus true positive rate, and the area under this curve indicates how well the saliency map predicts actual DME diagnoses. The two distributions are exactly equal when AUC is equal to 1, not relative when AUC is equal to 0.5, and negatively relative when AUC is equal to 0.

EMD. It represents the minimum cost of change of a distribution to another distribution. In this study, we use the fast implementation of EMD provided by Pele and Werman [27, 28]. EMD equal to zero means the two distributions are identical; a larger EMD means the two distributions are more different.

SS. It is another metric for measuring the similarities of two distributions. It first normalizes two distributions to let the sum equal one and then to sum the minimum values in each position. SS is always between zero and one. SS equal to one means two distributions are identical and SS equal to zero means two distributions are totally different.

Then, three quality measurements, classical sensitivity, specificity, and Youden's J statistic, were computed. The sensitivity and specificity were calculated for the whole set of classified pixels, that is, whether or not a pixel belonged to a RoI. Classically, the performance of a method is well described using sensitivity and specificity; they account for the individual result of hits or misses. However, we are

Table 1: Performance comparison of nine models in the EDMERI dataset.

Metrics	GT	Ours	AIM	AWS	GBVS	ITTI	STB	Judd	SUN	Torralba	Average
AUC	1.0000	0.8275	0.5610	0.6081	0.6748	0.6419	0.4723	0.7945	0.6449	0.6235	0.6498
EMD	0.0000	8.1524	13.4767	13.2513	12.3116	12.8744	17.9996	12.3079	12.4320	12.701	12.8341
SS	1.0000	0.1930	0.0853	0.0987	0.1066	0.1044	0.0055	0.1033	0.0932	0.0988	0.0991

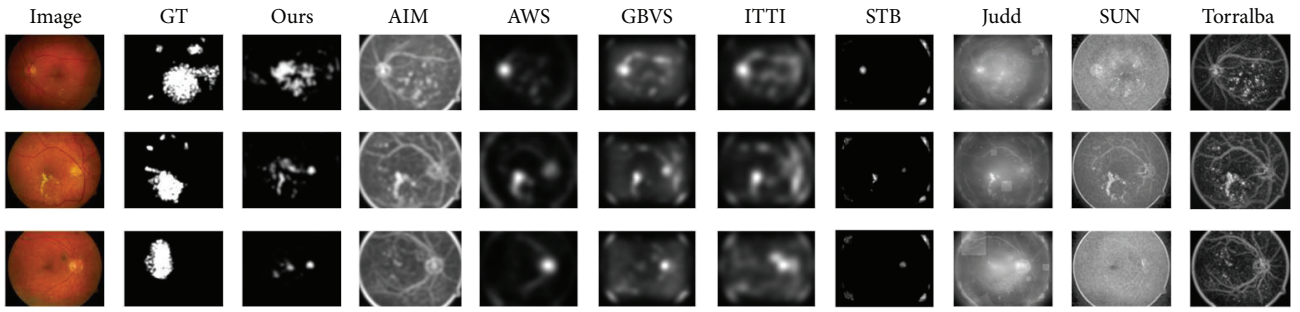


Figure 3: Some saliency maps produced by 9 different models from the EDMERI database along with predictions of several models using ROC. Each example shown by one row. From left to right: original image, ground truth, Ours, AIM, AWS, GBVS, ITTI, STB, Judd, SUN, and Torralba. It is obvious that Ours is more similar to the ground truth than other saliency maps.

interested in finding regions of interest, that is, collections of pixels with semantic meaning. Hence, the numbers of regions found by each method were also compared and the sensitivity of each method, regarding the number of RoIs, was also calculated.

Sensitivity. It is also called the true positive rate, or the recall in some fields, which measures the proportion of positives which are correctly identified, and is complementary to the false negative rate. The higher the sensitivity is, the more sensitive the diagnostic test is.

Specificity. It is also called the true negative rate, which measures the proportion of negatives which are correctly identified, and is complementary to the false positive rate. The higher the specificity is, the more precise the diagnostic test is.

Youden's J Statistic. It is also called Youden's index; this can be written as formula (4). Its value ranges from 0 to 1. The index gives equal weight to false positive and false negative values. The higher Youden's index is, the higher the authenticity the test has is. Consider

$$\text{Youden's index} = \text{sensitivity} + \text{specificity} - 1. \quad (4)$$

4.1. DME ROIs Prediction

4.1.1. Analysis of AUC, EMD, and SS. As far as we know, this is the first investigation devoted to extraction of DME RoIs information from retinal images, using a bioinspired model. The developed method was compared with eight well-known techniques which had to deal with similar challenges, but in natural scene. We used them as the baseline because they also emulate the visual system, even though they are not specifically devised to detect relevancy in medical images; these eight models were AIM [29], AWS [14], Judd [17], ITTI [5], GBVS [15], SUN [5], STB [30], and Torralba [4]. We trained and tested our model over the dataset following 10-fold cross validation. For EDMERI, $M = 10$. The statistical results are shown in Table 1.

Table 1 shows the comparison of evaluation performances of the 9 models in the EDMERI database. In this experiment, the average values of 10 times 10-fold cross validation in Table 1 are used for comparison. In the results, Ours has the best value in AUC, EMD, and SS. The AUC of our model is highest (0.8275), followed by Judd (0.7945). However, the average is only 0.6498. And the lowest value of EMD is shown in our model (8.1524), which is less than the average 12.8341. It means the results of Ours are more identical with ground truth than other models. Ours also has the best performance in SS with a value of 0.1930. The average value of SS is 0.0991, which just approximates half of Ours. Generally speaking, Ours has good performance in these three metrics. And Figure 3 presents some examples of the saliency maps produced by our approach and the other eight saliency models.

In Figure 4, we see the ROC curves of three examples in Figure 3 describing the performance of different saliency models. The size of the salient region plays an important role in the ROC method. Since it cannot separate salient regions from backgrounds using a certain threshold, the ROC method treats a saliency map as a binary classifier for ground truth under various thresholds. As shown in Figure 4, ROC of Ours was higher than other models from the 5% to 20% salient region; GBVS, Judd, and ITTI2 got higher ROC when the salient region was larger than 60%. This means that when the definition of the salient region changes, the rank of performance using the ROC method may change. However, the salient DME RoIs in a retinal image region are generally

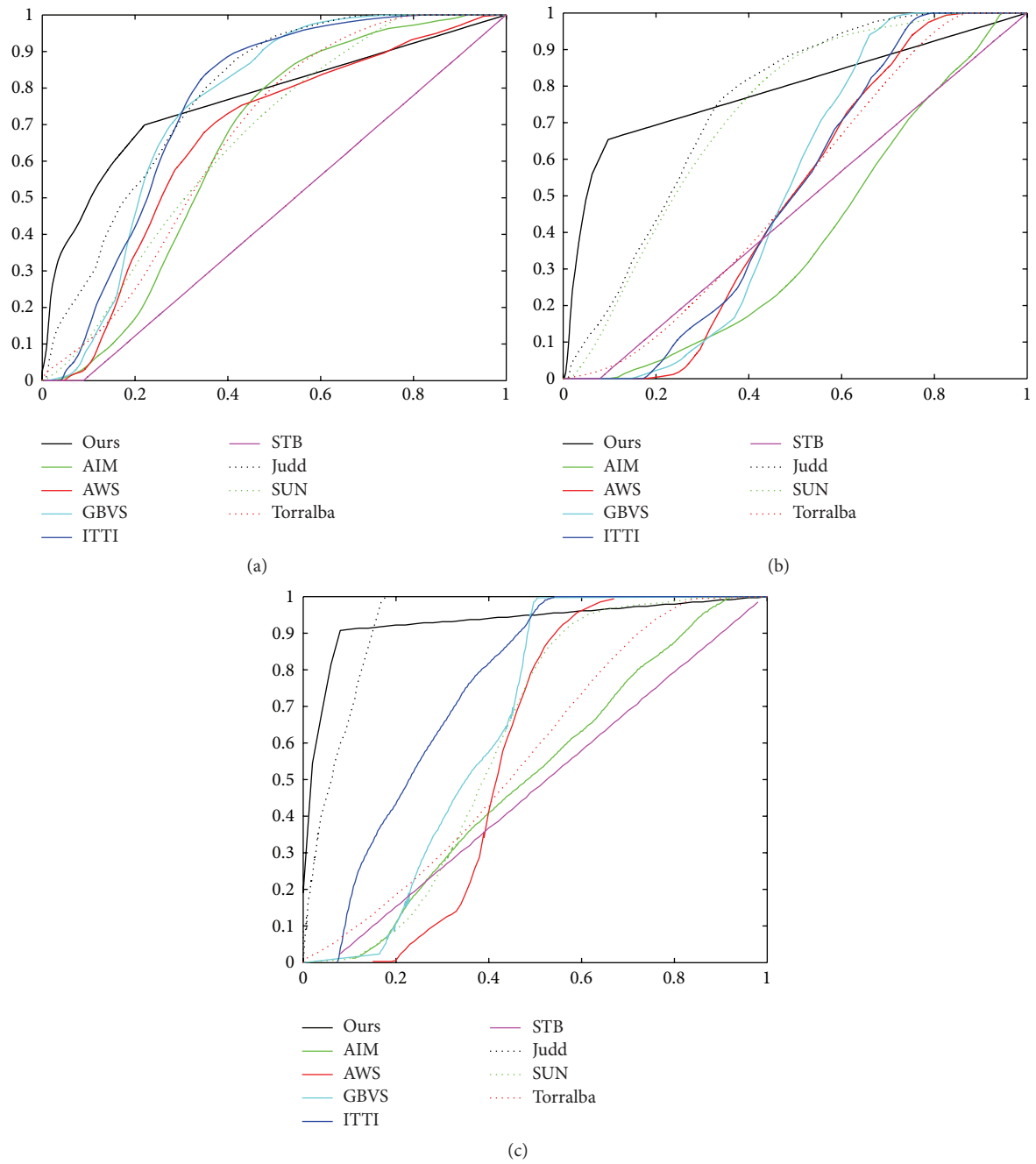


Figure 4: Some ROC curves produced by 9 different models from the three examples in Figure 3. ROC of Ours was higher than other models from the 5% to 20% salient region; GBVS, Judd, and ITTI2 got higher ROC when the salient region was larger than 60%. The salient DME RoIs in a retinal image region are generally under 20%. As a result, the performance of our method is better than other models when the salient region is small.

under 20% and even smaller in many cases. As a result, the performance of our method is better than other models when the salient region is small.

4.1.2. Analysis of Sensitivity and Specificity. The ability of the different methods to extract DME RoIs information from retinal images was evaluated using conventional sensitivity and specificity measurements. These results are shown in Table 2.

Table 2 shows sensitivities and specificities of the 9 models in 50% salient region. Overall, all sensitivity, specificity, and Youden measurements evidence that our model outperforms the other models. The sensitivity of our model is 83.7%, which surpasses the average sensitivity 18.2%, followed by

Table 2: Sensitivities and specificities of nine models.

	Ours	AIM	AWS	GBVS	ITTI	STB	Judd	SUN	Torralba	Average
Sensitivity (%)	83.7	50.7	57.0	80.1	69.6	81.6	37.2	65.6	63.7	65.5
Specificity (%)	77.7	59.2	64.5	57.0	58.1	74.1	60.1	60.0	55.4	62.9
Youden	0.614	0.099	0.215	0.371	0.277	0.557	−0.027	0.256	0.191	0.356

STB with 81.6% and GBVS with 80.1%. However, Judd had the lowest rate (only 37.2%), less than half of Ours. And the larger value of specificity (77.7%) is also shown in our model, which exceeds the average specificity 14.8%. Although the sensitivities of STB and GBVS are over 80%, their specificities are 74.1% and 57.0%, respectively; both are under Ours. The sensitivity of GBVS especially is even lower than the average. Owing to having the highest value of sensitivity and specificity, Youden's index (0.614) of our model is the highest among the 9 models, followed by STB with 0.557 and GBVS with 0.371. Average Youden's index is 0.356, which is only higher than half of Ours. The indisputable fact is that the higher Youden's index is, the higher the authenticity the test has is, and our model outperforms the other models in all sensitivity, specificity, and Youden measurements based on Table 2. Thus, Ours is suitable for extracting DME RoIs information from retinal images.

4.2. DME ROIs Detection. Almost all salient region detection approaches utilize a saliency operator, where from there they start to segment the most salient object. Because they are not specifically devised to detect relevancy in medical images, there is little study investigating the relevance of computational saliency models in medical images in the context of abnormality detection. Here, we used three well-known techniques as the baseline and showed that our approach could provide a good such starting point; these three models were ITTI [3], SR [19], and Achanta's [31].

We calculate ROC curves in Figure 5 by binarizing the saliency map using every possible fixed threshold, similar to the fixed thresholding experiment in [31]. As seen from the comparison (Figure 5), our saliency model performs better while competing with the other three state-of-the-art models tailored for this task. Figure 6 shows examples with diagnosis and detections of our model and the other three salient region detection models. As can be seen, our model is able to successfully detect the DME ROIs, ITTI's RoIs, and SR's RoIs mismatching the ground truth, and even worse, Achanta's cannot detect the DME ROIs.

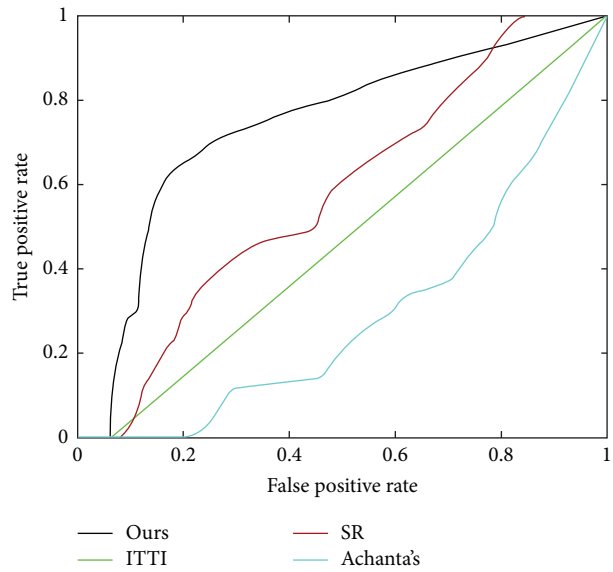


Figure 5: ROC curves for comparison of our model with ITTI, SR, and Achanta's.

5. Discussions and Conclusions

The present paper has introduced a novel strategy, a new visual saliency model using the Bayesian probability theory and machine learning techniques, for selecting DME RoIs in retinal images. The model is inspired in the first phase of a DME pathological examination, a process largely studied which starts by scanning the retinal images.

So far the underlying mechanism that controls a DME RoIs selection in retinal image has been poorly studied. Recent studies suggest that some visual mechanisms, such as the one that allows highlighting an object from the background and the visual attentional process, are connected. This fact suggests that the visual system is able to selectively focus on specific areas of the image, which besides are entailed with a high relevant meaning. Yet the idea is far from being fully exploited; our approach has been able to capture some basic facts; that is to say, that relevancy is a global property somehow constructed by integrating local features.

The proposed strategy is based on the interaction of Position Property and Feature Property and combined by a simple intersection operation using the Bayesian probability theory and machine learning techniques to obtain saliency maps. Our model is unlike traditional contrast-based bottom-up methods in that its learning mechanism has the ability to automatically learn the relationship between saliency and features. Moreover, unlike existing learning-based models that only consider the components of features themselves, our model simultaneously considers appearing frequency of features and the pixel location of features, which intuitively have a strong influence on saliency. As a result, our model can determine saliency regions and detect DME ROIs more precisely. Experimental results indicate that the proposed model has significantly better performance than other state-of-the-art models.

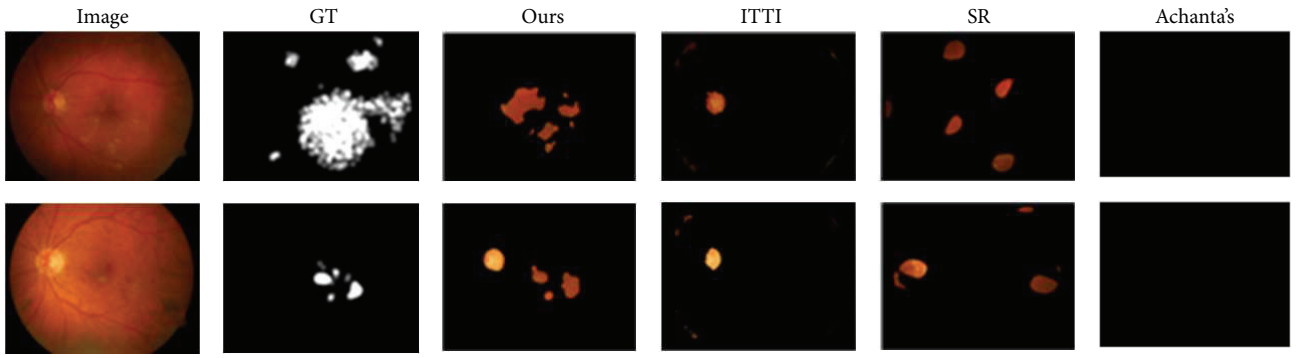


Figure 6: Some unnormalized saliency map for DME ROIs detection produced by 4 different models from the EDMERI database. Each example shown by one row. From left to right: original image, GT, Ours, ITTI, SR, and Achanta's. It is obvious that Ours is more similar to the ground truth than other saliency maps.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work is supported by the NSF of China (nos. 61117115 and 61201319), the Fundamental Research Funds for the Central Universities, and NWPU "Soaring Star" and "New Talent and Direction" Program.

References

[1] A. D. Fleming, K. A. Goatman, S. Philip et al., "The role of haemorrhage and exudate detection in automated grading of diabetic retinopathy," *British Journal of Ophthalmology*, vol. 94, no. 6, pp. 706–711, 2010.

[2] Early Treatment Diabetic Retinopathy Study Research Group, "Treatment techniques and clinical guidelines for photocoagulation of diabetic macular edema: early treatment diabetic retinopathy study report number 2," *Ophthalmology*, vol. 94, no. 7, pp. 761–774, 1987.

[3] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[4] A. Torralba, "Modeling global scene factors in attention," *Journal of the Optical Society of America*, vol. 20, no. 7, pp. 1407–1418, 2003.

[5] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: a bayesian framework for saliency using natural statistics," *Journal of Vision*, vol. 8, no. 7, pp. 1–20, 2008.

[6] DIARETDB: Diabetic Retinopathy Database and Evaluation Protocol, http://www.it.lut.fi/project/imageret/diaretdb0/index.html#DOWNLOAD.

[7] R. P. Phillips, T. Spencer, P. G. B. Ross, P. F. Sharp, and J. V. Forrester, "Quantification of diabetic maculopathy by digital imaging of the fundus," *Eye*, vol. 5, no. 1, pp. 130–137, 1991.

[8] A. Hunter, J. A. Lowell, B. Ryder, A. Basu, and D. Steel, "Automated diagnosis of referable maculopathy in diabetic retinopathy screening," in *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '11)*, pp. 3375–3378, IEEE, Boston, Mass, USA, September 2011.

[9] K. S. Deepak and J. Sivaswamy, "Automatic assessment of macular edema from color retinal images," *IEEE Transactions on Medical Imaging*, vol. 31, no. 3, pp. 766–776, 2012.

[10] A. Osareh, M. Mirmehdi, B. Thomas, and R. Markham, "Automatic recognition of exudative maculopathy using fuzzy C-means clustering and neural networks," in *Proceedings of the Medical Image Understanding and Analysis Conference (MIUA '01)*, pp. 49–52, Birmingham, UK, 2001.

[11] S. T. Lim, W. M. D. W. Zaki, A. Hussain, S. L. Lim, and S. Kusalavan, "Automatic classification of diabetic macular edema in digital fundus images," in *Proceedings of the IEEE Colloquium on Humanities, Science and Engineering (CHUSER '11)*, pp. 265–269, Penang, Malaysia, December 2011.

[12] G. P. Pena and J. S. Andrade-Filho, "How does a pathologist make a diagnosis?" *Archives of Pathology and Laboratory Medicine*, vol. 133, no. 1, pp. 124–132, 2009.

[13] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, no. 4, pp. 219–227, 1985.

[14] A. G. Diaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil, "Decorrelation and distinctiveness provide with human-like saliency," in *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pp. 343–354, Springer, Berlin, Germany, 2009.

[15] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS '06)*, pp. 545–552, Vancouver, Canada, December 2006.

[16] W. Kienzle, A. F. Wichmann, B. Scholkopf, M. O. Franz, and B. Schölkopf, "A nonparametric approach to bottom-up visual saliency," in *Conference on Neural Information Processing Systems (NIPS '07)*, pp. 689–696, 2007.

[17] J. Tilke, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proceedings of the 12th International Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 2106–2113, IEEE, Kyoto, Japan, October 2009.

[18] V. Jampani, Ujjwal, J. Sivaswamy, and V. Vaidya, "Assessment of computational visual attention models on medical images," in *Proceedings of the 8th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP '12)*, ACM, December 2012.

[19] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR '07), pp. 1–8, IEEE, Minneapolis, Minn, USA, June 2007.

[20] R. Gutiérrez, F. Gómez, L. Roa-Peña, and E. Romero, "A supervised visual model for finding regions of interest in basal cell carcinoma images," *Diagnostic Pathology*, vol. 6, article 26, pp. 97–105, 2011.

[21] MESSIDOR: Methods to evaluate segmentation and indexing, http://messidor.crihan.fr/download-en.php.

[22] STARE: Structured Analysis of the Retina, http://messidor.crihan.fr/download-en.php.

[23] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[24] G. Kootstra, A. Nederveen, and B. de Boer, "Paying attention to symmetry," in *Proceedings of the 19th British Machine Vision Conference (BMVC '08)*, pp. 1115–1125, September 2008.

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 2, 2011.

[26] N. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Advances in Neural Information Processing Systems (NIPS '05)*, pp. 298–308, 2005.

[27] O. Pele and M. Werman, "A linear time histogram metric for improved SIFT matching," in *Proceedings of the 10th European Conference on Computer Vision (ECCV '08)*, vol. 5304, pp. 495–508, Marseille, France, October 2008.

[28] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV '09)*, pp. 460–467, Kyoto, Japan, October 2009.

[29] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Advances in Neural Information Processing Systems*, pp. 155–162, MIT Press, 2006.

[30] W. Dirk and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

[31] R. Achantay, S. Hemamiz, F. Estraday, and S. Süsstrunky, "Frequency-tuned salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1597–1604, IEEE, Miami, Fla, USA, June 2009.

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2015, Article ID 653639, 10 pages
http://dx.doi.org/10.1155/2015/653639

Research Article

Real-Time Control of a Video Game Using Eye Movements and Two Temporal EEG Sensors

Abdelkader Nasreddine Belkacem,[1,2] Supat Saetia,[3] Kalanyu Zintus-art,[3] Duk Shin,[4] Hiroyuki Kambara,[4] Natsue Yoshimura,[4] Nasreddine Berrached,[2] and Yasuharu Koike[4,5]

[1]*Department of Neurosurgery, Osaka University Medical School, Osaka 565-0871, Japan*
[2]*Intelligent Systems Research Laboratory, University of Sciences and Technology of Oran, 00031 Oran, Algeria*
[3]*Department of Information Processing, Tokyo Institute of Technology, Yokohama 226-8503, Japan*
[4]*Precision and Intelligence Laboratory, Tokyo Institute of Technology, Yokohama 226-8503, Japan*
[5]*Solution Science Research Laboratory, Tokyo Institute of Technology, Yokohama 226-8503, Japan*

Correspondence should be addressed to Abdelkader Nasreddine Belkacem; belkacem011@hotmail.com

Received 18 May 2015; Revised 9 July 2015; Accepted 30 July 2015

Academic Editor: Pietro Arico

EEG-controlled gaming applications range widely from strictly medical to completely nonmedical applications. Games can provide not only entertainment but also strong motivation for practicing, thereby achieving better control with rehabilitation system. In this paper we present real-time control of video game with eye movements for asynchronous and noninvasive communication system using two temporal EEG sensors. We used wavelets to detect the instance of eye movement and time-series characteristics to distinguish between six classes of eye movement. A control interface was developed to test the proposed algorithm in real-time experiments with opened and closed eyes. Using visual feedback, a mean classification accuracy of 77.3% was obtained for control with six commands. And a mean classification accuracy of 80.2% was obtained using auditory feedback for control with five commands. The algorithm was then applied for controlling direction and speed of character movement in two-dimensional video game. Results showed that the proposed algorithm had an efficient response speed and timing with a bit rate of 30 bits/min, demonstrating its efficacy and robustness in real-time control.

1. Introduction

Electroencephalogram (EEG) is a noninvasive technique for measuring electrical potentials from electrodes placed on the scalp produced by brain activity and some other artifacts such as Electrooculogram (EOG) and Electromyogram (EMG). Nowadays, EEG technique has been used to establish portable synchronous and asynchronous brain-computer interfaces (BCIs). Noninvasive EEG-based BCIs are the most promising interface for space applications. They can be classified as "evoked" or "spontaneous." An evoked BCI exploits a strong characteristic of the EEG, the so-called evoked potential, which reflects the immediate automatic responses of the brain to some external stimuli. Spontaneous BCIs are based on the analysis of EEG phenomena associated with various aspects of brain function related to mental tasks carried out by the subject at his/her own will.

BCIs offer people with movement disabilities a means of interaction with their environment by translating brain activity into device control [1]. Recently, several BCIs have been developed based on evoked potentials such as P300 and steady-state visual evoked potential (SSVEP) or based on slow potential shifts and variations of rhythmic activity [2]. Many critical issues are faced on the development of a BCI such as classification accuracy, number of degrees of freedom, and training process (i.e., how users learn to operate the BCI). Some researchers have demonstrated that BCI users can learn to control their brain activity through video games [3, 4]. Therefore, EEG-controlled gaming applications can provide strong motivation for practicing. In this respect, a main issue

Table 1: Comparison table of EOG- and video-based eye-tracking techniques.

Criteria	EOG electrodes	Video-based eye tracking
Intrusiveness	Intrusive with electrodes attached to the face (i.e., electrodes mounted on the skin around the eye).	Intrusive for cameras attached to glasses; nonintrusive for cameras mounted independently.
Complexity	(i) Electrodes number reduces the portability of the technique (many electrodes attached on the face). (ii) EOG is a simple and easy method to measure eye movements and it is still commonly used clinically for testing eye movements in patients.	(i) The algorithm complexity of the image processing system. (ii) Calibration is a crucial problem: head distance, head and pupil range of rotation with respect to sagittal plane of the body must be estimated (in some case manual corrections are still needed).
Influence of noise	Facial muscles (EMG signal) can be influenced on EOG signal.	(i) Light: big problem for image processing. (ii) Head movement: must keep your eyes open and in the vision field of the camera. (iii) Hard to use it in real-life application (outside environment).
Processing time	Fast: training or calibration phase needed.	Long: training or calibration phase needed; image processing takes much memory.
Classification accuracy	High, but related to visual angle, number of electrodes, and algorithm applied.	High, but related to head angle, user environment, and algorithm applied.

is how to develop medical and nonmedical games to improve the robustness of BCIs with the goal of making it a more practical and reliable technology.

Eye movements and blink artifacts are pervasive problems in EEG-based BCI research [5]. However, the present authors feel that these artifacts are actually a valuable source of information and are useful for communication and control. In this paper, several participants were tested in different real-time experiments on different days to examine the variability and nonstationary nature of EEG signals. This study has shown that the same control performances can be obtained via either EOG or EEG signals with using suitable positions and minimum number of sensors for EEG technique. The control performances of participants were tested in natural environment where they were asked to perform the movements of their eyes, body, and head as naturally as possible.

In Section 2, BCI-based medical and nonmedical games, popular techniques for eye tracking, and hybrid BCIs based on brain activity and eye movement are reviewed. In Section 3, materials and methods for developing several paradigms of real-time experiment are introduced. These experiments are based on real-time classification and control with opened and closed eyes using our proposed algorithm with minimum number of EEG sensors. In Section 4, results of eye movements' classification and video game control are presented. Advantages and disadvantages of the proposed idea in different scenarios are discussed with detailed aspects in Section 5. Finally, conclusion and prospects of future work are given in Section 6.

2. Related Work

2.1. BCI-Based Games. Simple BCI-based games can help inexperienced users control via brain activity. Games based on EEG have been designed to increase the intensity or duration of attention, increase the speed and accuracy of brain-signal control, and improve other capabilities [2]. Two types of BCI-based games are frequently seen: medical and nonmedical games. For medical purposes, Lalor et al. [3] describe a game intended to improve the concentration needed to operate a BCI that uses SSVEP. Other medical games were designed to encourage rapid generation of motor imagery-based BCI commands and enhance the user's experience [4]. For entertainment purposes, several BCI-based games were developed, for example, the one based on popular video game "Tetris", playing pinball, and dancing robot [6–8]. Most of these nonmedical games are based on concentration level of the player [8].

2.2. Eye Tracking. In daily life conversation, eye movements play an important role in interaction with environment by indicating a person's direction and level of attention. Fortunately, most of handicapped people can still control their eyes. Thus, eye movement can be an additional option to improve their quality of life. Eye movements can be measured as EOG signals or via cameras and applied to communication or control systems [9–19]. Both methods have their respective merits and demerits (Table 1).

2.3. Hybrid BCIs. Recent studies have shown that EOG signals acquired using EEG technique with a minimal number of EEG sensors around the frontal lobe or ears are practical to detect and classify eye movements [20–23]. Therefore, brain activity was not extracted from EEG to be used as additional information. Hybrid BCIs offer a potentially effective control for complex systems through the combination of brain- and non-brain-based activities. Wheelchair control, for example, requires multiple degrees of freedom and fast intention detection, making solely EEG-based wheelchair control a challenge. Each type of BCIs has its limitations, but a hybrid BCI combines different approaches to utilize the advantages of multiple modalities [24, 25].

A hybrid BCI combining motor imagery and P300 was proposed in Li et al. [26]. It was further used to control the

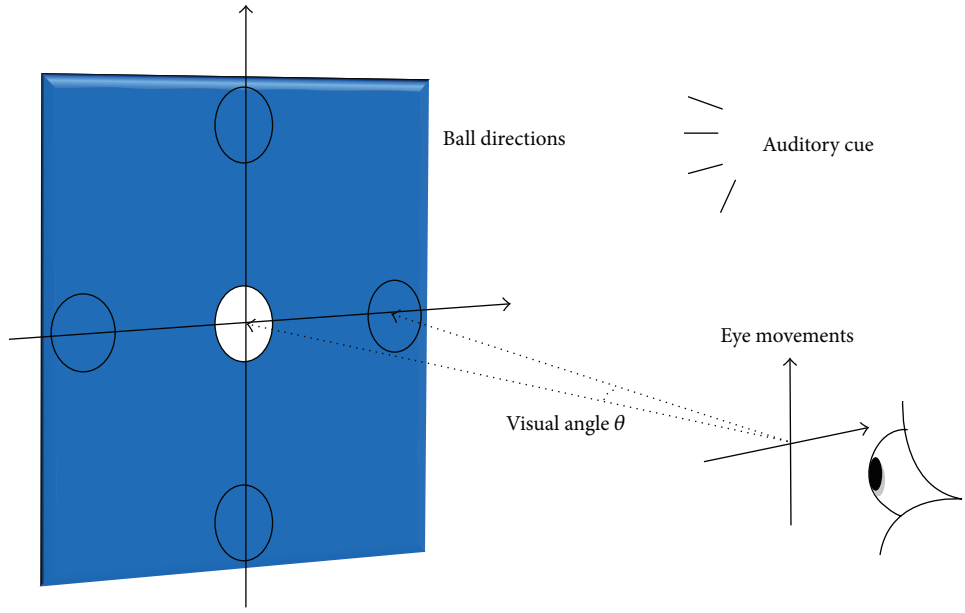


Figure 1: Real-time experiment for controlling a white ball with opened and closed eyes based on eye movement.

direction and speed of a wheelchair in Long et al. [27]. However, a fast and accurate design for the stop command and the forward and backward commands has not been obtained. Wang et al. [28] proposed asynchronous wheelchair control with a hybrid EEG-EOG BCI, combining motor imagery, P300 potentials, and eye blinking. Their experimental results not only demonstrated the efficiency and robustness of brain-controlled wheelchair systems but also indicated that the participants could control the wheelchair spontaneously and efficiently without any other assistance. However, Wang et al. used only a single eye movement, eye-blinks. Here, we show that more than six classes of eye movements can be classified and used for real-time control, demonstrating the utility of EOG signals in EEG data. Hereafter we provide explanations of the experiment paradigm, EEG recording, and real-time video game control, describe and discuss our classification and control results, and then conclude with future prospects.

3. Materials and Methods

3.1. Experimental Paradigm. Five participants (4 males, 1 female) with a mean age of 26.2 years (standard deviation (SD): 2.5) were seated in a chair and instructed to watch a monitor screen located in different positions away at eye level. All subjects reported normal or corrected-to-normal vision and had no prior BCI experience. One of them suffers from Amblyopia (vision problem also called lazy eye). Subjects participated in a real-time test experiment, followed by an eye-controlled video game. The real-time test experiment was created to evaluate the performance of the proposed algorithm. Participants then played the video game using eye movements. In this study, classification accuracy was calculated using the real-time test experiment, and control performance was evaluated using the video game.

In the real-time test experiment (Figure 1), five participants were asked to move a white ball to five positions (up, down, left, right, and center) using eye movements or change its color by blinking. The participants did not move their eyes on consistent time interval during control period. Subjects performed ten runs (10 trials for each eye movement), with each run lasting 60 s. During the first 10 s, the participants were asked to fixate a white ball in the center. Then they were asked to move the white ball to one of the four cardinal directions (up, down, right, or left) using eye movements. In the last 10 s, the participants were asked to blink three times to change the color of the ball from white to yellow. The participants were asked then to move a white ball to five positions with closed eyes using voice instructions to show the feasibility of sending commands in real time by blind persons for autonomous eye movement based control systems. After testing the performances of the proposed algorithm in this real-time experiment, the participants were able to play a video game during 20 minutes using eye movements without any training or calibration phase. In real-time control of video game, the subjects can move their head position and direction and watched the motion of both a game character and meteors in various timings. Figure 2 shows the experiment framework and electrode positions for eye-controlled gaming.

3.2. EEG Recording. EEG signals were acquired during real-time experiments using a g.USBamp system (g.tec medical engineering, Austria) at a sampling frequency of 256 Hz. Two EEG electrodes were applied on the upper area behind the left and right ears (Figure 2). This proposed position was favorable because it was not obstructed by hair and allowed for capture of EOG without the discomfort that might occur with electrodes on the face. A clip electrode was attached to the right earlobe as reference and ground

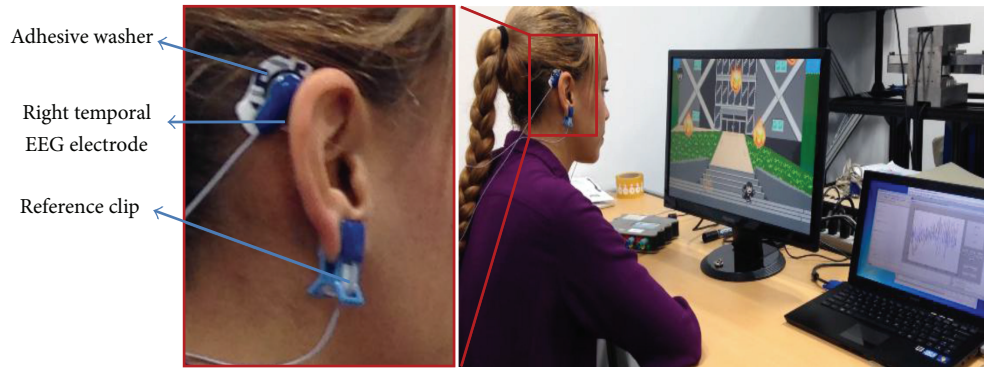


Figure 2: Setup for EEG recording and game control.

electrode was placed on the forehead. To prevent muscle artifacts, participants were asked to avoid strong blinking and head movements.

An 8th-order Butterworth band-pass filter with a lower cut-off frequency of 0.5 Hz and an upper cut-off frequency of 100 Hz was applied to the recorded signals. A 4th-order 48–52 Hz notch filter was used to suppress 50 Hz power-line noise.

3.3. Classification Algorithm. After recording the EEG signals from right and left hemispheres, a real-time algorithm was applied to distinguish between six classes: blink, center, right, left, up, and down. Signal data were sent from the amplifier to the computer in 1 s blocks. The EEG signals were separated into low and high frequency components to separate EOG activity from brain activity (1). Thus, 4th-order Butterworth high and low pass filters with cut-off of 10 Hz were used to decompose EEG signals into two frequency bands: low [0.5–10 Hz] and high [10–100 Hz]. For preprocessing phase, the baseline artifact was corrected by subtracting the smoothed signal with its mean (2). In the current study, EOG signal and eye-blink artifact included in observed EEG signal were used as valuable sources of information:

$$\begin{aligned} \mathrm{EEG}(t)_{\mathrm{Observed}} &= \mathrm{EEG}(t)_{\mathrm{source}} + \mathrm{EOG}(t) + \mathrm{EMG}(t) \\ &\quad + \mathrm{Artifacts}, \end{aligned} \tag{1}$$

$$\begin{aligned} X_i &= \left(\mathrm{EEG}_i - \mu\left(\mathrm{EEG}(nT)\right)\right), \\ \mathrm{EEG}_i &= \mathrm{EEG}(iT), \\ \mu\left(\mathrm{EEG}(nT)\right) &= \sum_{i=1}^{n} \frac{\mathrm{EEG}(iT)}{n}, \end{aligned} \tag{2}$$

where $1/T$ is sampling frequency of the EEG signal ($t = nT$, $n = 1, 2, \ldots, 256$). After baseline correction, two signals Y_1 and Y_2 were calculated (3). Y_1 maximized the margin between classes left and right by using the difference between the left ($X_{L,\mathrm{EEG}}$) and right ($X_{R,\mathrm{EEG}}$) electrode signals. Y_2 distinguished between classes up and down using the smoothed sum of the two electrodes. A real-time detection was added before classification phase because the length of time interval of eye movements was not fixed. The length of time interval

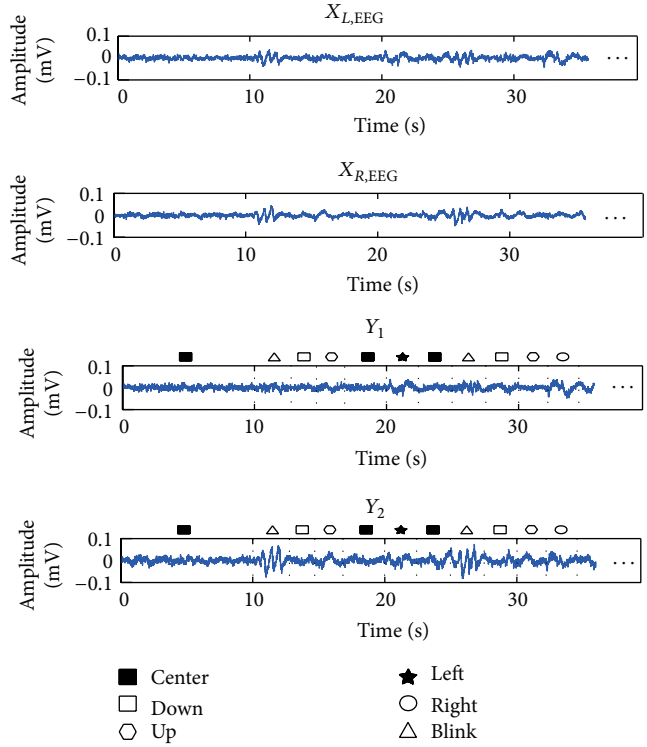


Figure 3: Example of raw EEG signals $X_{L,\mathrm{EEG}}$ and $X_{R,\mathrm{EEG}}$, from the left and right electrodes, respectively, and processed signals Y_1 and Y_2. The symbols represent blink and eye movement classes.

was varying depending on each trial of natural eye movement and control timing for each participant (Figure 3):

$$\begin{aligned} Y_1 &= X_{L,\mathrm{EEG}} - X_{R,\mathrm{EEG}}, \\ Y_2 &= X_{L,\mathrm{EEG}} + X_{R,\mathrm{EEG}}. \end{aligned} \tag{3}$$

In our previous work [20], we tested an offline algorithm for classifying between eye movements in four cardinal directions using area under the curve for signals Y_1 and Y_2. Then we added features for online discrimination between six classes of eye movements [21]. Signals corresponding to each eye orientation have a specific shape (Figure 3), and the blink

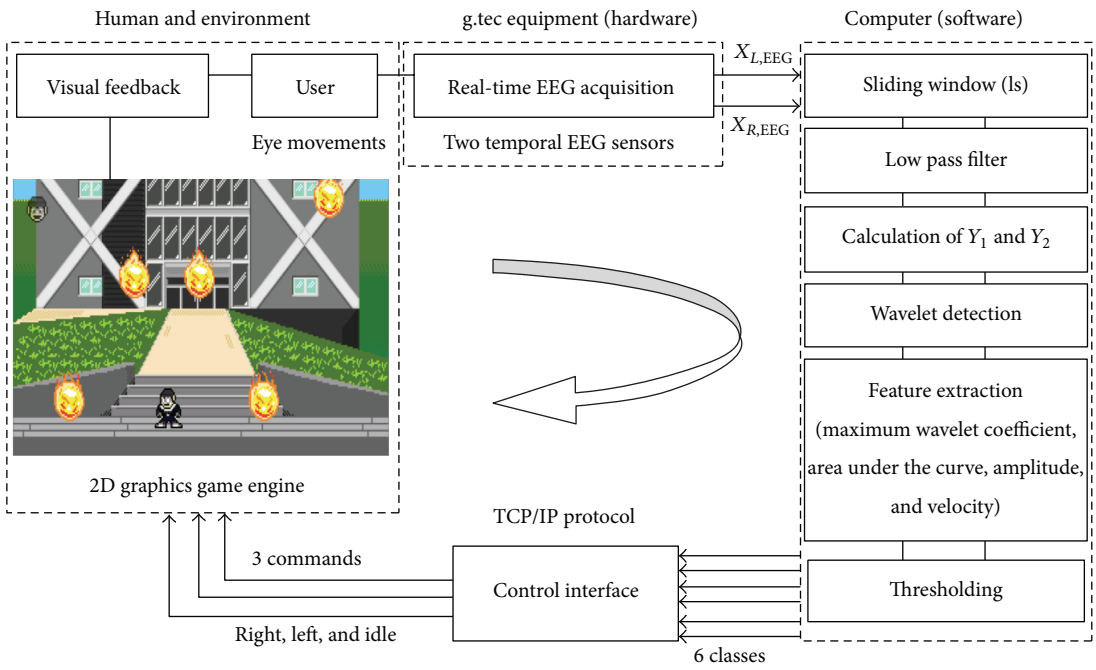


Figure 4: Control flowchart for the real-time eye-controlled game.

pulse is similar to a Gaussian pulse. The wavelet scalogram was used for detection phase of eye movement. We applied a continuous wavelet transform on signals Y_1 and Y_2. For a scale parameter, $a > 0$, and position, b, the CWT is

$$\mathrm{EEG}_{a,b}(\omega) = C_{a,b} = \int_0^{1\,\mathrm{s}} \mathrm{EEG}(t)\,\psi_{a,b}(t)\,dt,$$
$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi^*\left(\frac{t-b}{a}\right), \tag{4}$$

where $C_{a,b}$ is the continuous wavelet transform coefficients, a is positive and defines the scale, b is any real number and defines the shift, $\psi_{a,b}(t)$ is a wavelet function of Haar, $*$ denotes the complex conjugate, and EEG(t) is Y_1 or Y_2 signal. CWT is a real-valued function of scale and position because the signal EEG(t) is real-valued. By continuously varying the values of the scale parameter, a, and the position parameter, b, we obtained the wavelet coefficients. Then the wavelet scalogram was obtained by computing:

$$S = |\mathrm{coefs} * \mathrm{coefs}|,$$
$$E = \sum S(:), \tag{5}$$

where coefs is the CWT coefficients, S is the energy for each wavelet coefficient, and E is the energy of the wavelet coefficients for 1 s. For both Y_1 and Y_2 signals, area under the curve of the positive and negative peaks was used for feature extraction. The area was calculated over a 200-ms window centered on the position of maximum wavelet coefficient using the trapezoidal method. The waves were situated in between the maximum wavelet coefficient.

Hierarchical classification was used to discriminate between patterns obtained from Y_1 and Y_2 signals. Fixed thresholds for four features, maximum wavelet coefficient, area under the curve, amplitude, and velocity, were set prior to the real-time experiments by calculating their means and standard deviations based on our previous studies [20, 21]. The classification results were converted into vectors to produce binary outputs. These outputs were used to move a cursor on a screen and control a video game.

3.4. Eye-Controlled Game. To demonstrate the efficacy of eye-based control in real world applications, we created a simple game. The game was an obstacle evasion 2D platformer game. In the game, a character had to run in two directions to avoid being hit by falling meteors which appeared in semirandom sequences on a closed stage. The character's movement was controlled with the player's eye movements (Figure 4).

The implementation of the game was divided into two modules, the EEG signal classification module and the graphic game module. The EEG signal classification module was implemented in MATLAB (MathWorks, Natick, MA), the graphic game module was implemented using the Unity 4 game development ecosystem (Unity Technologies, San Francisco, CA), and the game's logic was written in C#. The two modules interfaced with each other via TCP/IP protocol.

3.5. Character's Movement Mechanism. The classification module was capable of discriminating between 6 eye movement classes (left, right, center, up, down, and blink). However, the video game required only 3 commands: "left," "right," and "idle," so the left and right classes were mapped to their respective commands and the remaining classes were mapped to the "idle" command. The character's direction and speed were defined by the sign and magnitude of a unit vector, respectively. Initially, the vector was set to 0, so the character

Table 2: Vocabulary of real-time commands for eye-controlled gaming.

	EEG signal	Character action
Command 1	Eyes moving to the up position and then returning back	Stop
Command 2	Eyes moving to the down position and then returning back	Stop
Command 3	Eyes moving to the left position and then returning back	Move at the left side
Command 4	Eyes moving to the right position and then returning back	Move at the right side
Command 5	Blinking	Stop
Command 6	No eye movement (fixation)	Stop
Command 7	Two successive similar movements of eyes to the left or right direction	Increase the speed
Command 8	Two successive opposite movements of eyes such as moving to the left then right position or vice versa	Decrease the speed

stood still when the game started. A positive vector made the character move to the right, and a negative vector made it move to the left. Left commands from the classification module decreased the vector magnitude by 0.1 units, while right commands increase the magnitude by 0.1 units. The character's speed changed by increasing or decreasing a fixed magnitude for character's acceleration. Commands were received discretely every 1 s to control the character's motion continuously. So the character's movement direction was determined as the dominant command in a given command sequence window. This approach was used to compensate the natural sudden change in eye movement direction when the eyes move back to the center position. The idle command was different from the movement commands in that the vector magnitude was set to 0 immediately after the idle command was received, allowing the character to stop when the player intended with no delay. The maximum speed of the meteors was defined by a vector of units 0.1, and the initial speed was 0. Acceleration downwards was defined as a vector of units 0.01. The number of meteors was semirandomized, but no more than 5 appeared at one time. There were 5 meteors release points lined up evenly across the top of the screen (Figure 4). Each point had its own set of semirandomized release times and delay values. We set the values to release, on average, 3 meteors per repetition. Difficulty was controlled by the number of meteors and their speed. Therefore, classification accuracy was based on stopping and moving the character and not on avoidance of the meteors. Table 2 summarizes the actions to be performed by the character shown in Figure 3 corresponding to each eye movement.

3.6. Evaluation Criteria. The performance of the proposed algorithm was evaluated in the real-time experiments by calculating the classification accuracy for six and three classes based on success or failure to move the ball or character to a desired direction, respectively. For control of the video game, precision, sensitivity, and specificity were calculated for three classes (right, left, and idle) such that

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100, \\ \text{Sensitivity (Recall)} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100, \\ \text{Specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100, \end{aligned} \tag{6}$$

Table 3: Confusion matrix of the six classes and accuracies (rounded %) averaged across all participants.

	Up	Down	Right	Left	Center	Blink
Up	**42**	14	0	2	28	14
Down	6	**50**	0	0	24	20
Right	0	0	**96**	4	0	0
Left	0	0	0	**100**	0	0
Center	4	0	0	2	**88**	6
Blink	0	0	6	4	2	**88**

Table 4: Confusion matrix of the five classes and accuracies (rounded %) averaged across all participants with closed eyes.

	Up	Down	Right	Left	Center
Up	**65**	5	0	10	20
Down	12	**46**	19	15	8
Right	0	0	**98**	2	0
Left	0	0	2	**97**	1
Center	4	1	0	0	**95**

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

4. Results

Calculating Y_1 and Y_2 resulted in unique signatures for each class (Figure 3) which could be exploited for classification. Table 3 shows a confusion matrix of the six classes and accuracies averaged across five participants (16.67% chance level). All participants demonstrated reliable control of the white ball in the first experiment, achieving an average accuracy of 77.33 (SD: 2.52%). The proposed algorithm showed high classification accuracy for right, left, center, and blink classes using 0% of the data for training or calibration phase and 100% of the data for testing.

We used a one-way ANOVA to evaluate real-time classification results of the six eye movement classes among participants. No significant differences were observed between participants for accuracy ($F(4, 25) = 0.06$, $P = 0.993$).

Table 4 shows a confusion matrix of the five classes (up, down, left, right, and center) and accuracies averaged across five participants with closed eyes using the same

threshold values. All participants resulted in almost the same classification accuracy with opened or closed eyes, achieving an average accuracy of 80.2% (SD: 1.87%) using auditory feedback with closed eyes (20% chance level) with no significant difference between participants ($F(4, 20) = 0.08$, $P = 0.989$).

We observed that classification accuracies using closed eyes of "center" and "up" classes were increased compared to the case of using opened eyes. This accuracy difference between up class in the case of "opened eyes" and "closed eyes" is due to similarity of the wave shape of "up direction" and "eye-blink" signals. The "up" and "blink" classes were combined in one class in eye-closed experiment. Therefore, the classification accuracy of up class was increased. The accuracy of center class was also increased because there is no noise related to the visual information. The algorithm showed a high accuracy and robustness using a single trial for controlling the white ball in real time in the opened- and closed-eyes situations with no significant difference between all users using the same thresholds values for all of them. The participant with Amblyopia disease showed the lowest classification accuracy compared to others due to the difficulty in moving his eyes correctly.

In real-time control of video game, the subjects can move their eyes position and direction and watched the motion of game character and meteors in the various timings. Table 5 shows precision, sensitivity, and specificity for each participant. These values were calculated based on accuracies (Table 2), with up, down, center, and blink classes considered as the idle class. Average sensitivity was over 90%, and participant 3 achieved an accuracy of around 100% using a single trial to make decision. No significant differences were found between participants for accuracy ($F(4, 10) = 1.23$, $P = 0.359$), sensitivity ($F(4, 10) = 0.63$, $P = 0.653$), or specificity ($F(4, 10) = 0.94$, $P = 0.478$).

Response speed and timing are also important in full control of a BCI. Using serial communication, the classification algorithm processed 60 bits/min, but the control algorithm processed the first bit and ignored the second. Therefore, the bit rate for controlling the video game was 30 bits/min. The proposed algorithm was useful in classification accuracy and time-saving because the main problem faced by real-time application is the computing and processing time.

Table 5: Precision, sensitivity, and specificity values (rounded %) for each participant during real-time game play.

	Right	Idle	Left
Participant 1 (M)	100/90/100	95/100/100	70/100/94
Participant 2 (M)	83.3/100/95.9	92.5/97.4/100	90.9/100/97.9
Participant 3 (M)	100/100/100	100/100/100	100/100/100
Participant 4 (F)	90.9/100/98	95/100/95.3	90.9/100/98
Participant 5 (M)	100/90.9/100	100/100/100	90.9/100/97.5

5. Discussion

In this study, subjects were able to perform real-time control of an interface using six eye movements and play a video game with three eye movement based commands. Because the resting position of human eyes is forward-facing, we return our eyes back to the center position after looking at any other direction. This action would have resulted in classifications opposite to the intended direction and, in turn, adversely affect interface control. To solve this problem, the game character's movements did not follow the commands sent from classification module verbatim. The movement of the character was defined as a unit vector of acceleration along x-axis, with "right" being positive values and "left" being negative values. Movement commands gradually increased the acceleration value in the intended direction. This technique reduced the effect of the eyes returning to the center position. For the "idle" or stop command, which required an immediate response, the movement vector magnitude was immediately returned to zero to stop the character stop as soon as the player intended.

For classes up and down, even though the two sensors were located at the same points behind the right and left ears, we were able to obtain discriminable EOG activity. We believe that the eyes did not move at mirrored angles across the central axis. This dissimilarity likely made detection of up and down directions possible and was amplified by calculating Y_2. Table 6 summarizes the advantages and disadvantages of the proposed robust real-time control system based on EEG signal.

Since the magnitude of the electrical signal generated by the eye movement depends on the angular velocity, many researchers have used a big visual angle of between 30° and 45° to get a high accuracy for detecting the directions or positions of the eye movement [12, 29–33]. This large visual angle is not suitable for daily life applications because it leads almost immediately to eye fatigue, exhausting the user. Comparing the real-time results using opened eyes from this study with those of our previous offline and online classification work [20, 21], we found that classification accuracy using a small visual angle decreased from almost 90% to 77%. This was likely due to the complexity of the real-time application, environment conditions, and users' behavior. The participants were asked to make themselves comfortable and perform the movements as naturally as possible. There were instances where eye movements were misclassified, but the signal data showed no serious influence by head or body movements. Although future versions of the proposed algorithm would benefit from an automatic thresholding subroutine instead of a calibration phase, results showed that the current algorithm holds promise in real-time applications.

Through this work, we can help not only handicapped people but also the blind persons to use their eye movements using voice commands with auditory feedback for controlling smart-home applications. For able-bodied users, the idea of sending commands with closed eyes can decrease the fatigue issue related to rich detailed visual environments. In some special eye movement based applications, the visual information can be replaced by information from the tactile, olfactory, or auditory senses such as the case of reducing or increasing the room temperature and the volume of music.

Table 6: Advantages and disadvantages of eye movements classification based on EEG signal.

Criteria	Advantage	Limitation
Visual angle	A small visual angle between 5° and 10° was used to decrease fatigue issue (a large visual angle of 30° or more is required to detect eye movement in most research using EOG signals. This large visual angle leads almost immediately to eye fatigue, exhausting the user).	It becomes difficult to detect eye movements if the visual angle is less than 5°.
User	Several participants were tested (offline [20], online [21], and in different real-time experiments in this study) on different days to examine the variability and nonstationary nature of EEG signals.	Absence of testing the proposed algorithm on handicapped users.
Sensors position & number	(i) The position of sensors around the ears is more robust to muscles activity noise (body or head movements do not influence so much the classification accuracy). (ii) Two temporal EEG sensors were used (4 attached sensors on the face are used as minimum requirement to get good classification accuracy in EOG technique).	A low-cost wireless device based on the proposed idea is not yet developed.
Comfort and portability	The most suitable sensors position for daily life applications to record eye movements compared with EOG sensors (the sensors can be attached to the end of the glasses arms (temples), headset, and headband).	Less comfort [21].
Real-time classification	(i) Single trial was used for real-time classification. (ii) No training or calibration phase was added before real-time classification (fixed and common thresholds for all subjects were used). (iii) No fixed time interval for eye movements (the user is free to move his/her eyes and send commands at any moment). (iv) Six classes were distinguished using a linear clarifier. (v) Eye movements were detected and classified in open- and closed-eyes cases. (vi) The proposed algorithm was tested in several real-time scenarios.	Using average or loop to make a decision or machine learning methods can improve the classification accuracy but decrease the response time [9–13, 29].
Real-time control	(i) Asynchronous control (the user can send commands even with closed eyes using noninvasive technique). (ii) The classification results were used for full control of continuous character's movement in 2D video game. (iii) The bit rate for controlling the video game was 30 bits/min.	For each application, we need to develop an interface between classification results and the controlled device.
Classification accuracy	Classification accuracy with chance level of 16.67% was greater than 70%, the suggested minimum for reliable BCI control with chance level of 50% [34].	As same as EOG technique [20].

We sought to contribute to the development of noninvasive, asynchronous, and hybrid BCIs combining brain activity and eye movements. This kind of BCI could offer utility in daily life applications and practical machine control. Though most approaches in the BCI field focused solely on brain activity, we see an opportunity for advancement of this field by combining EEG and EOG. This approach could be used to assist both able-bodied and disabled persons with high efficiency compared to existing BCIs.

6. Conclusions

This paper presented asynchronous and robust real-time control of a video game through eye movements detected using two temporal EEG sensors. The algorithm was designed for multiclass control in a visually elaborate immersive 2D game. Results of the study indicated that successful multiclass control is possible using suitable position of sensors to detect and classify eye movements in opened- and closed-eyes situations.

In the near future, for rehabilitation, medical therapy, and entertainment, we would like to design portable, non-invasive, and asynchronous hybrid EEG-EOG-based games and smart-home applications using minimum number of wearable wireless sensors.

Conflict of Interests

The authors declare that there is no conflict of interests.

Authors' Contribution

The work presented here was carried out in collaboration between all authors, but the main part has been accomplished by the first author. Abdelkader Nasreddine Belkacem made real-time experiments, developed the proposed algorithm, and analyzed the data; Supat Saetia and Kalanyu Zintus-art designed 2D graphics of the video game; Duk Shin, Hiroyuki Kambara, Natsue Yoshimura, Nasreddine Berrached, and

Yasuharu Koike supervised the presented work, edited the final version of the paper, and approved it.

Acknowledgments

This work was supported by the Japan Intentional Cooperation Agency (JICA) and "Development of BMI Technologies for Clinical Application" carried out under the Strategic Research Program for Brain Sciences by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[2] J. R. Wolpaw and E. W. Wolpaw, *Brain-Computer Interface: Principles and Practice*, Oxford University Press, New York, NY, USA, 1st edition, 2012.

[3] E. C. Lalor, S. P. Kelly, C. Finucane et al., "Steady-state VEP-based brain-computer interface control in an immersive 3D gaming environment," *Eurasip Journal on Applied Signal Processing*, vol. 2005, no. 19, pp. 3156–3164, 2005.

[4] D. Marshall, D. Coyle, S. Wilson, and M. Callaghan, "Games, gameplay, and BCI: the state of the art," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 5, no. 2, pp. 82–99, 2013.

[5] B. Noureddin, P. D. Lawrence, and G. E. Birch, "Online removal of eye movement and blink EEG artifacts using a high-speed eye tracker," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 8, pp. 2103–2110, 2012.

[6] A. Finke, A. Lenhardt, and H. Ritter, "The Mindgame: a P300-based brain-computer interface game," *Neural Networks*, vol. 22, no. 9, pp. 1329–1333, 2009.

[7] M. Tangermann, M. Krauledat, K. Grzeska et al., "Playing pinball with non-invasive BCI," *Advances in Neural Information Processing Systems*, vol. 21, pp. 1641–1648, 2009.

[8] Q. Wang, O. Sourina, and M. K. Nguyen, "Fractal dimension based neurofeedback in serious games," *Visual Computer*, vol. 27, no. 4, pp. 299–309, 2011.

[9] M. I. Rusydi, M. Sasaki, and S. Ito, "Affine transform to reform pixel coordinates of EOG signals for controlling robot manipulators using gaze motions," *Sensors*, vol. 14, no. 6, pp. 10107–10123, 2014.

[10] H. Zeng, A. Song, R. Yan, and H. Qin, "EOG artifact correction from EEG recording using stationary subspace analysis and empirical mode decomposition," *Sensors*, vol. 13, no. 11, pp. 14839–14859, 2013.

[11] R. Barea, L. Boquete, S. Ortega, E. López, and J. M. Rodríguez-Ascariz, "EOG-based eye movements codification for human computer interaction," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2677–2683, 2012.

[12] A. Güven and S. Kara, "Classification of electro-oculogram signals using artificial neural network," *Expert Systems with Applications*, vol. 31, no. 1, pp. 199–205, 2006.

[13] C.-C. Postelnicu, F. Girbacia, and D. Talaba, "EOG-based visual navigation interface development," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10857–10866, 2012.

[14] D. Borghetti, A. Bruni, M. Fabbrini, L. Murri, and F. Sartucci, "A low-cost interface for control of computer functions by means of eye movements," *Computers in Biology and Medicine*, vol. 37, no. 12, pp. 1765–1770, 2007.

[15] N. Itakura and K. Sakamoto, "A new method for calculating eye movement displacement from AC coupled electro-oculographic signals in head mounted eye-gaze input interfaces," *Biomedical Signal Processing and Control*, vol. 5, no. 2, pp. 142–146, 2010.

[16] L. Y. Deng, C.-L. Hsu, T.-C. Lin, J.-S. Tuan, and S.-M. Chang, "EOG-based Human-Computer Interface system development," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3337–3343, 2010.

[17] N. Timmins and M. F. Marmor, "Studies on the stability of the clinical electro-oculogram," *Documenta Ophthalmologica*, vol. 81, no. 2, pp. 163–171, 1992.

[18] V. Häkkinen, K. Hirvonen, J. Hasan et al., "The effect of small differences in electrode position on EOG signals: application to vigilance studies," *Electroencephalography and Clinical Neurophysiology*, vol. 86, no. 4, pp. 294–300, 1993.

[19] W. W. Abbott and A. A. Faisal, "Ultra-low-cost 3D gaze estimation: an intuitive high information throughput compliment to direct brain–machine interfaces," *Journal of Neural Engineering*, vol. 9, no. 4, Article ID 046016, 2012.

[20] A. N. Belkacem, H. Hirose, N. Yoshimura, D. Shin, and Y. Koike, "Classification of four eye directions from EEG signals for eye-movement-based communication systems," *Journal of Medical and Biological Engineering*, vol. 34, no. 6, pp. 581–588, 2014.

[21] A. N. Belkacem, D. Shin, H. Kambara, N. Yoshimura, and Y. Koike, "Online classification algorithm for eye-movement-based communication systems using two temporal EEG sensors," *Biomedical Signal Processing and Control*, vol. 16, pp. 40–47, 2015.

[22] H. Manabe, M. Fukumoto, and T. Yagi, "Conductive rubber electrodes for earphone-based eye gesture input interface," in *Proceedings of the 17th ACM International Symposium on Wearable Computers (ISWC '13)*, pp. 33–39, ACM, New York, NY, USA, September 2013.

[23] H. Manabe and M. Fukumoto, "Full-time wearable headphone-type gaze detector," in *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, pp. 1073–1078, ACM, New York, NY, USA, 2006.

[24] G. Pfurtscheller, T. Solis-Escalante, R. Ortner, P. Linortner, and G. R. Muller-Putz, "Self-paced operation of an SSVEP-based orthosis with and without an imagery-based "brain switch": a feasibility study towards a hybrid BCI," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 4, pp. 409–414, 2010.

[25] B. Z. Allison, C. Brunner, V. Kaiser, G. R. Müller-Putz, C. Neuper, and G. Pfurtscheller, "Toward a hybrid brain-computer interface based on imagined movement and visual attention," *Journal of Neural Engineering*, vol. 7, no. 2, Article ID 026007, 2010.

[26] Y. Li, J. Long, T. Yu et al., "An EEG-based BCI system for 2-D cursor control by combining Mu/Beta rhythm and P300 potential," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 10, pp. 2495–2505, 2010.

[27] J. Long, Y. Li, H. Wang, T. Yu, J. Pan, and F. Li, "A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 5, pp. 720–729, 2012.

[28] H. Wang, Y. Li, J. Long, T. Yu, and Z. Gu, "An asynchronous wheelchair control by hybrid EEG-EOG brain-computer interface," *Cognitive Neurodynamics*, vol. 8, no. 5, pp. 399–409, 2014.

[29] D. Kumar and E. Poole, "Classification of EOG for human computer interface," in *Proceedings of the 2nd Joint Engineering in Medicine and Biology Conference, 24th Annual International Conference of the Engineering in Medicine and Biology Society (EMBS/BMES '02)*, vol. 1, pp. 64–67, Houston, Tex, USA, 2002.

[30] R. Barea, L. Boquete, M. Mazo, and E. López, "System for assisted mobility using eye movements based on electrooculography," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 4, pp. 209–218, 2002.

[31] A. B. Usakli and S. Gurkan, "Design of a novel efficient human-computer interface: an electrooculagram based virtual keyboard," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2099–2108, 2010.

[32] J.-H. Yu, B.-H. Lee, and D.-H. Kim, "EOG based eye movement measure of visual fatigue caused by 2D and 3D displays," in *Proceedings of the IEEE-EMBS International Conference on Biomedical & Health Informatics*, pp. 305–308, January 2012.

[33] K. Yamagishi, J. Hori, and M. Miyakawa, "Development of EOG-based communication system controlled by eight-directional eye movements," in *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '06)*, pp. 2574–2577, New York, NY, USA, August 2006.

[34] G. Pfurtscheller, C. Neuper, and N. Birbaumer, "Human brain-computer interface," in *Motor Cortex in Voluntary Movements: A Distributed System for Distributed Functions*, E. Vaadia and A. Riehle, Eds., Methods and New Frontiers in Neuroscience, pp. 367–401, CRC Press, 2005.