Applied Soft Computing 12 (2012) 2072–2084

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Low cost remote gaze gesture recognition in real time

David Rozado *,[1], Francisco B. Rodriguez [1], Pablo Varona [1]

Escuela Politécnica Superior, Calle Francisco Tomás y Valiente, 11, Universidad Autónoma de Madrid, Madrid 28049, Spain

ARTICLE INFO

Article history:
Received 12 May 2011
Received in revised form 13 February 2012
Accepted 28 February 2012
Available online 21 March 2012

Keywords:
Encoding
Motion analysis
Multidimensional sequences
Multidimensional signal processing
Neural network architecture
Patternrecognition
Gaze tracking
Human computer interaction

ABSTRACT

Predefined sequences of eye movements, or 'gaze gestures', can be consciously performed by humans and monitored non-invasively using remote video oculography. Gaze gestures hold great potential in human–computer interaction, HCI, as long as they can be easily assimilated by potential users, monitored using low cost gaze tracking equipment and machine learning algorithms are able to distinguish the spatio-temporal structure of intentional gaze gestures from typical gaze activity performed during standard HCI. In this work, an evaluation of the performance of a bioinspired Bayesian pattern recognition algorithm known as Hierarchical Temporal Memory (HTM) on the real time recognition of gaze gestures is carried out through a user study. To improve the performance of traditional HTM during real time recognition, an extension of the algorithm is proposed in order to adapt HTM to the temporal structure of gaze gestures. The extension consists of an additional top node in the HTM topology that stores and compares sequences of input data by sequence alignment using dynamic programming. The spatio-temporal codification of a gesture in a sequence serves the purpose of handling the temporal evolution of gaze gestures instances. The extended HTM allows for reliable discrimination of intentional gaze gestures from otherwise standard human–machine gaze interaction reaching up to 98% recognition accuracy for a data set of 10 categories of gaze gestures, acceptable completion speeds and a low rate of false positives during standard gaze–computer interaction. These positive results despite the low cost hardware employed supports the notion of using gaze gestures as a new HCI paradigm for the fields of accessibility and interaction with smartphones, tablets, projected displays and traditional desktop computers.

1. Introduction

The emergence of tablets and smartphones with touch sensitive surfaces as their only input modality has spurred a growing interest in the subject of gestures in human–computer interaction or HCI [1–4]. In this work, the feasibility of gaze gestures as an input modality to control a computer by carrying out sequences of gaze positions is explored. Video oculography is used to track eye movements and gestures are recognized in real time with a Bayesian bio-inspired pattern recognition algorithm known as Hierarchical Temporal Memory (HTM).

A gaze gesture can be defined as a sequence of strokes. A stroke is an intentional movement between two fixation points. Different patterns of strokes define different gestures [5]. Different gestures can then be mapped to issue different commands for HCI purposes.

The use of eye movements as a pointing mechanism to control an interface has been extensively studied in the literature [6] but the usage of gaze gestures in HCI is a recent concept. Gaze tracking systems using video-based hardware and algorithms can determine the Point of Regard (PoR), where the user is looking at, on a screen [7,8]. Usually, the PoR is employed as a pointing device, that is, as a substitute of the mouse [9]. Although eye movements can reveal intention to interact with an object, the eyes lack an activation mechanism, and therefore anywhere a user looks, it is not clear for the interaction system whether it should issue a control command or not [10]. Discerning whether the user looks at an object to examine it or to interact with it is known as the Midas touch problem [9] and it highlights the need for additional forms of interaction through gaze. Ideally, selection should be performed with an external switch, decoupling pointing and selection, in the same way as it is done with a mouse. However, people with severe disabilities or users in certain environments are often unable to use an external selection device, and must therefore rely on gaze-only activation techniques. Among these, blinking and dwell-time activations are the most common. In the former, a click command is issued when the system detects a blink with a predefined duration, while in the latter the click command is issued when a fixation longer than a predefined threshold is detected. Gaze gestures emerge as a potential candidate to bridge the gap between pointing and selection in gaze interaction systems by circumventing the Midas touch problem.

Due to the novelty of the concept of gaze gestures, the amount of research done on the subject is not very extensive. Authors in

* Corresponding author. Tel.: +34 914972361; fax: +34 914972235.
E-mail address: david.rozado@uam.es (D. Rozado).
[1] All authors belong to the GNB group at the Escuela Politécnica Superior, Universidad Autónoma de Madrid.

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2012.02.023

[11] used gaze gestures in a dialog system to facilitate communication but the subjects involved in the experiment were not trying to consciously perform gaze gestures. Research has been done on the topic of using gaze gestures for gaze-based input of characters [12–14]. The work from [5] explores the interaction possibilities of single stroke gaze gestures. The works [1,15] are extensive studies on the subject of gaze gestures, both in terms of possibilities and limitations of the technology. In particular, [15], generates an optimal data set of gaze gestures using the less probable paths users are likely to perform during gaze interaction. In [14], authors attempt to employ gaze gestures to issue commands and enter characters. In [12] and [13], experienced users enter characters through gaze gestures at up to 7.99 words per minute. The study from [16] demonstrated that a small set of symbols is preferable in gaze gesture based operating interfaces.

In this article, it is shown how gaze gestures constitute an emerging and viable paradigm for HCI. Gaze gestures can be employed by people with severe disabilities, who use gaze as a mono-modal input in their HCI [17]. Additionally, gaze gestures, when used in combination with other input devices, can provide an additional input channel that augments and enhances the interaction possibilities with a computer [18]. This multi-modal interaction paradigm would not only benefit people with disabilities, but it could also provide a new venue of interaction with small screen devices such as smartphones or tablets or with devices in environments where traditional interaction methods such as keyboard or mouse are either out of reach (e.g. media centers) or inconvenient to use (e.g. electronic devices in surgical rooms).

Gaze gestures differentiate themselves from control commands traditionally used in gaze–computer interaction such as fixations and dwell times. Due to the fast nature of the saccadic movements involved in gaze gestures, this selection technique can potentially be faster and less stressful than dwell time. Moreover, gaze gestures can also be very robust to inaccuracy problems and calibration shifts. However, there can be an overlap between natural search patterns and the gaze patterns of a gesture, which could lead to false positives, i.e. the accidental detection of an involuntary gaze gesture. For gaze interaction purposes, it is desirable to minimize unintended gaze gestures recognition [5]. Increasing the complexity of the gaze gesture, i.e. the number of strokes, minimizes the overlap between natural gaze patterns and consciously performed gaze gestures and increases the interaction vocabulary, but it also introduces a greater cognitive complexity and physiological load on the end user.

The potential of gaze gestures as a form of HCI relies heavily on the ability of machine learning algorithms to properly discriminate intentional gaze gestures from otherwise normal gaze activity during HCI. Gaze gestures can be described in terms of their spatio-temporal structure. The recognition of time series consisting of a spatio-temporal structure unfolding over time is a challenging pattern recognition problem, for which Recurrent Neural Networks and Hidden Markov Models are often used [19–21]. Spatio-temporal encoding of the features to be learned in a hierarchical system is an alternative and successful approach to solve this type of problem with low computational cost and robust performance. In this work, an extension of an existing pattern recognition algorithm, Hierarchical Temporal Memory (HTM), is presented to improve its performance in real time gaze gestures recognition.

HTM [22], is a conexionist pattern recognition paradigm inspired on neocortical principles of organization and function. HTM theory incorporates the hierarchical organization of the neocortex into its topology [23]. HTM uses spatio-temporal codification to encapsulate the structure of problems' spaces. Hierarchical organization and spatio-temporal coding are well documented principles for information processing in neural systems [23–25]. HTM algorithms perform robustly in traditional machine learning tasks such as image recognition [26] where patterns are represented as a complete set of spatial attributes. For problems where an instance is composed of time series of varying spatial arrangements, HTM performance is not as robust [27]. Hence, an extended HTM algorithm is suggested to improve traditional HTM performance on this type of problems. The proposed approach can also be applied to other types of problems whose instances also possess a temporal structure [28].

Gaze gestures are composed of a spatial shape that renders itself very suitable to the data structure that traditional HTMs are designed to encapsulate in their hierarchy. However, the temporal evolution of the gaze gesture pattern creates difficulties for traditional HTMs to properly separate the input space into categories. The extended Top Node and its inherent ability to warp time, expands the ability of HTMs to handle instances where sequences unfold over time at different speeds and with considerable levels of noise. Hence, the motivation for the modification of traditional HTMs lies on the limited accuracy of traditional HTM to solve the problem of gaze gesture recognition in real time. Note that in this work, the term "*real time*" will be used to refer to the user's perception of immediate recognition while carrying out gaze gestures in a manner that results transparent to the user and creates the illusion of 'real time' recognition while properly discriminating consciously performed gaze gestures from other types of gaze activity while interaction occurs with the computer. We refer the interested reader to the explanatory video [29] to illustrate the concept of real time gaze gesture recognition.

Even though eye tracking systems are prohibitively expensive for mainstream users (with prices ranging anywhere between 5000$ and 50,000$), in this work an open source eye tracking algorithm [6,30] and off-the shelf components are used to build the gaze tracking system. Hence, the gaze gesture recognition engine used in this work is extremely low cost and hence widely accessible (with the whole setup costing less than 50$) but it is also more noisy than commercial systems. Section 4 shows that gaze gesture recognition does not need expensive high-end hardware and that already built-in cameras on consumer devices such as smartphones or tablets could easily be used for gaze gesture recognition. The ability of the extended HTM system to flexibly learn a wide range of spatio-temporal patterns and its tolerance to noise make it specially appropriate to handle noisy gaze gestures recognition. The results demonstrate that the extended HTM algorithm can robustly separate natural eye movements during computer usage from intentional gaze gestures indicating that using gaze gestures recognized with the extended HTM algorithm constitutes an innovative, low cost, robust, easy-to-learn and viable approach to HCI for several environments and device combinations.

2. Methods

A gaze gesture recognition engine is conceptually placed between the gaze tracker and the application a user wishes to control. Fig. 1 illustrates this architecture. In the experiments, the performance of two modalities of gaze gestures are compared: those performed using dwell times at the beginning and at the end of the gesture to signal the gaze gesture boundaries and those performed without dwell time. Additionally, the performance of two recognition algorithms is compared: traditional HTM and extended HTM.

2.1. Eye tracking

A video-based gaze tracking system seeks to find the user's PoR using information obtained from the eye by one or more cameras that record the user's eye region. Infrared illumination can be

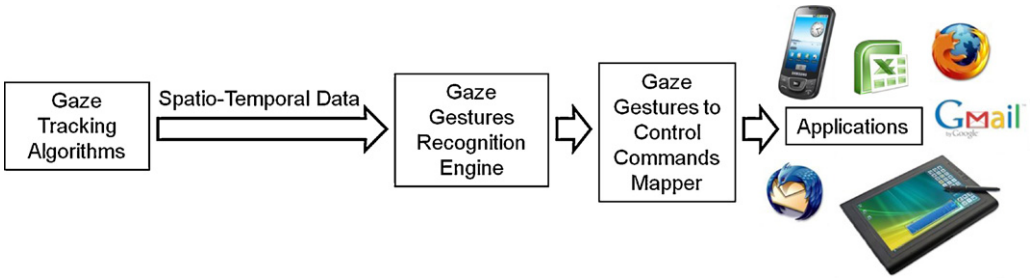

Fig. 1. System architecture. For application purposes, a gaze gesture recognition engine is conceptually placed between the gaze tracker engine and the applications that a user wishes to control. Intermediate communication interfaces are necessary to transmit gaze data, fetch it and map recognized gaze gestures to application-specific or universal control commands.

employed to improve image contrast and to produce a reflection on the cornea, known as corneal reflections or glints. These corneal reflections and the center of the pupil can be used to estimate the PoR. A calibration process consisting on users looking at several points on the screen is necessary. During calibration, several gaze samples per point on the screen are gathered by the gaze tracker engine and this data is used in a polynomial regression model to fit the gaze samples data. The training data used during the calibration process are the vector differences between the glint and the center of the pupil coordinates. The polynomial model is used to interpolate any vector differences between the glint and the center of the retina to specific regions in the screen. Since the infrared spectrum is invisible to the human eye, the infrared light that creates the glint in the cornea is not distracting nor annoying to the user. Fig. 2 shows a screenshot of an eye being tracked by the open-source ITU Gaze Tracker [6] with the center of the pupil and the corneal reflections clearly being identified and tracked.

2.2. Gaze gestures

Different conceptualizations of gaze gestures exist [17]. Gaze gestures can be relative, i.e. they can be performed anywhere on the screen, or absolute, requiring the user to direct gaze to a sequence of absolute positions on the screen. Due to the limited accuracy of eye tracking technology, fine discrimination between close points on the screen is often not possible. This and the fact that it is uncomfortable for users to accurately generate sequences of micro-movements, advocates the merits of performing gaze gestures by using absolute positions on the screen. Gaze gestures can also be classified as single stroke, when composed of just one saccadic movement, or complex, when they involve more elaborate paths

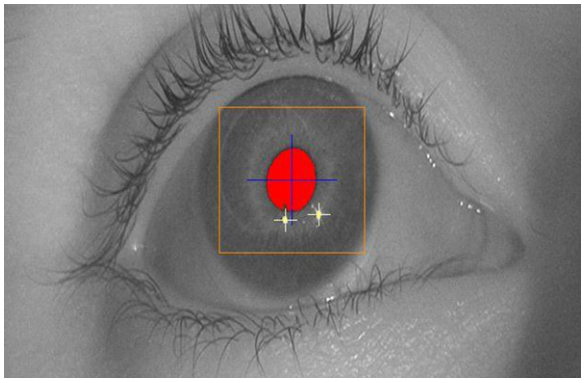

Fig. 2. The ITU gaze tracker tracking one eye. The gaze estimation algorithm of the open source ITU Gaze Tracker [6] used in this work tracks within the region of interest (yellow square) the pupil center (blue cross) and the corneal reflections (white crosses) created by the infrared light sources. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

[31]. The main advantage of simple gaze gestures is that they are easy to memorize and perform by users. Yet they markedly overlap with normal gaze activity when interacting with a computer, thus limiting their applicability as an input channel since normal inspection and navigation patterns might accidentally be confused with gaze gestures. Complex gaze gestures have the advantage of greatly increasing the vocabulary size of gaze interaction. However, complex gaze gestures generate a cognitive and physiological load on the user. Cognitively it is difficult for users to remember a large set of complex gestures, and physiologically it is tiring and challenging to complete them [5]. Finding the right trade-off between simple and complex gaze gestures is therefore paramount to successfully use gaze gestures as an input device. Furthermore, gaze gestures can be classified as saccadic gaze gestures, when the movements between fixation points are saccadic (ballistic) or gliding gaze gestures, where the gaze is glided along the whole trajectory of the gesture. In this work, absolute saccadic gaze gestures of intermediate complexity consisting on performing a sequence of saccades (strokes) between different areas of the screen are used.

2.3. HTM formalism

Traditional HTM technology uses a set of layers arranged in a hierarchy for topological organization. Each layer is composed of one or several computational nodes. Nodes are related through children-parent relationships. Each node throughout the hierarchy possesses an intrinsic receptive field formed by its children nodes or, in the case of bottom level nodes, a portion of the sensors' input space.

HTM nodes can function in two modes: training mode and inference mode. During training, nodes receive input vectors encapsulating the input space properties of the receptive field to which they are exposed. Nodes perform spatio-temporal codification on the input data by clustering input vectors into groups according to spatial and temporal adjacency. During inference, nodes emit binary output vectors indicating which cluster, from the set of clusters into which the training input vectors were partitioned, is active according to their similarity to the current input vector.

Nodes operate in discrete time steps and perform the same learning algorithm, differing only on the type of input vectors that they process. All nodes, except the top node, carry out unsupervised learning. The top node receives spatio-temporal information from children nodes, but it does not perform temporal aggregation of the data and it does not emit output vectors. The top node just maps incoming input vectors to the signaled category in a supervised fashion during training and infers through similarity measurements the proper category during inference.

HTM involves the use of a probabilistic generative model (see Fig. 3), and Bayesian belief propagation (see Fig. 4). Each node contains a set of coincidence patterns or CPs: $c_1, c_2, \ldots, c_n \in C$ and a

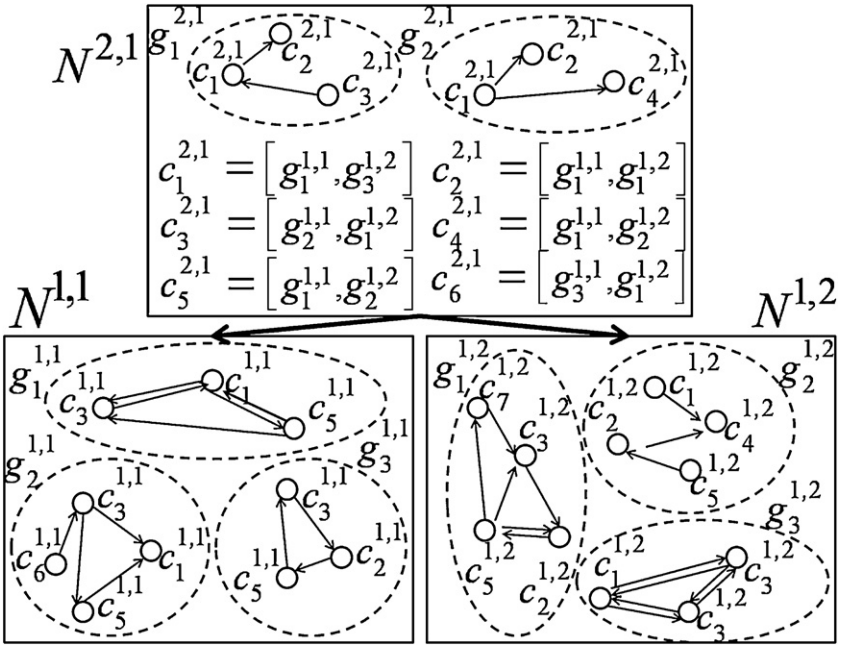

Fig. 3. HTM as a generative spatio-temporal model of data. A simple two level hierarchy consisting of a parent node and 2 children nodes is shown. Each node contains a set of CPs, *c*'s, and a set of MCs, *g*'s, defined over the set of CPs. A CP in a node represents a co-activation of a subset of MCs in its children nodes.

set of Markov chains or MCs: $g_1, g_2, \ldots, g_n \in G$. CPs represent co-occurrences of sequences from their children nodes. Each MC is defined as a subset of the set of CPs in a node. CPs capture the spatial structure of nodes or sensors underneath in the hierarchy by representing vectorially the co-activation of MCs in a node's children. A MC activated in a parent node concurrently activates its constituent MCs in the node's children. The MCs capture the temporal structure of a set of CPs, i.e., the likelihood of temporal transitions among them. The incoming vectors to an HTM node encapsulate the degree of certainty over the child MCs. With this information, the node calculates its own degree of certainty over its CPs. Based on the history of messages received, it also computes a degree of certainty in each of its MCs. This information is then passed to the parent node. Feedback information from parent nodes toward children nodes takes place by parent nodes sending to children nodes their degree of certainty over the children node's MCs, Fig. 3. A mathematical implementation of the described algorithm within the framework of belief propagation is presented in a formalized form in Fig. 4.

Numenta's Nupic package (v1.7.1) [22] which is an implementation of a traditional probabilistic HTM was used to run the experiments and for benchmarking purposes against the in-house developed extended HTM algorithm.

2.4. Extended HTM

Typical problems undertaken by HTMs [22] consist of instances whose spatial configuration is fully represented at any time instant, for example, image recognition [32]. Gaze gestures instead are composed of sequences of spatial arrangements over time that together constitute a sign. At any given time, *t*, the complete representation of the gesture is not available, just a particular spatial PoR of

Likelihood over CPs:	$y_t(i) = P(-e_t \mid c_i(t)) \propto \prod_{j=1}^{M} \lambda_t^{m_j}(r_i^{m_j})$ where CP c_i is the co-occurrence of $r_j^{m_1}$'th MC from child 1, $r_i^{m_2}$'th MC from child 2,..., and $r_i^{m_M}$'th MC from child M.
Feed-forward likelihood of MCs	$\lambda_t(g_r) = P(-e_0^t \mid g_r(t)) \propto \sum_{c_i(t) \in C^k} \alpha_t(c_i, g_r)$ $\alpha(c_i, g_r) = P(-e_t \mid c_i(t)) \sum_{c_j(t-1) \in C^k} P(c_i(t) \mid c_j(t-1), g_r) \alpha_{t-1}(c_j, g_r)$ $\alpha_0(c_i, g_r) = P(-e_0 \mid c_i(t=0)) P(c_i(t=0) \mid g_r)$
Belief distribution over CP	$Bel_t(c_i) \propto \sum_{g_r \in G^k} \beta_t(c_i, g_r)$ $\beta_t(c_i, g_r) = P(-e_t \mid c_i(t)) \sum_{e_j(t-1) \in C^k} P(c_i(t) \mid c_j(t-1), g_r) \beta_{t-1}(c_j, g_r)$ $\beta_0(c_i, g_r) = P(-e_0 \mid c_i(t=0)) P(c_i \mid g_r) \pi_0(g_r)$
Message sent to children nodes	$\pi^{m_i}(g_r) \propto \sum_i I(c_i) Bel(c_i)$, where $I(c_i) = \begin{cases} 1 & \text{if } g_r^{m_i} \text{ is even} \\ 0 & \text{otherwise} \end{cases}$

Fig. 4. Belief propagation equations for HTM nodes. The reader is encouraged to take the Node $N^{2,1}$ from Fig. 3 as reference. $N^{2,1}$ contains 6 CPs and two MCs. Each MC is composed of 3 CP. In this table, c_i is the *i*th coincidence in the node. g_r is the *r*th MC in the node. $-e_t$ indicates the bottom up evidence at instant *t*. $-e_0^t$ indicates the evidence sequence from time 0 ... *t*. $+e$ stands for top-down evidence. λ is the feed-forward output of the node. λ^{m_i} represents the feed-forward input to the node from its child node m_i. π is the feedback input to the node. π^{m_i} is feedback output of the node to its child node m_i. y is the bottom-up likelihood over CPs in a node. α is a bottom-up state variable for the MCs in a node. β is a state that combines bottom-up and top-down evidence for a MC in a node. B_{c_i} represents belief in the *i*th CP in a node.

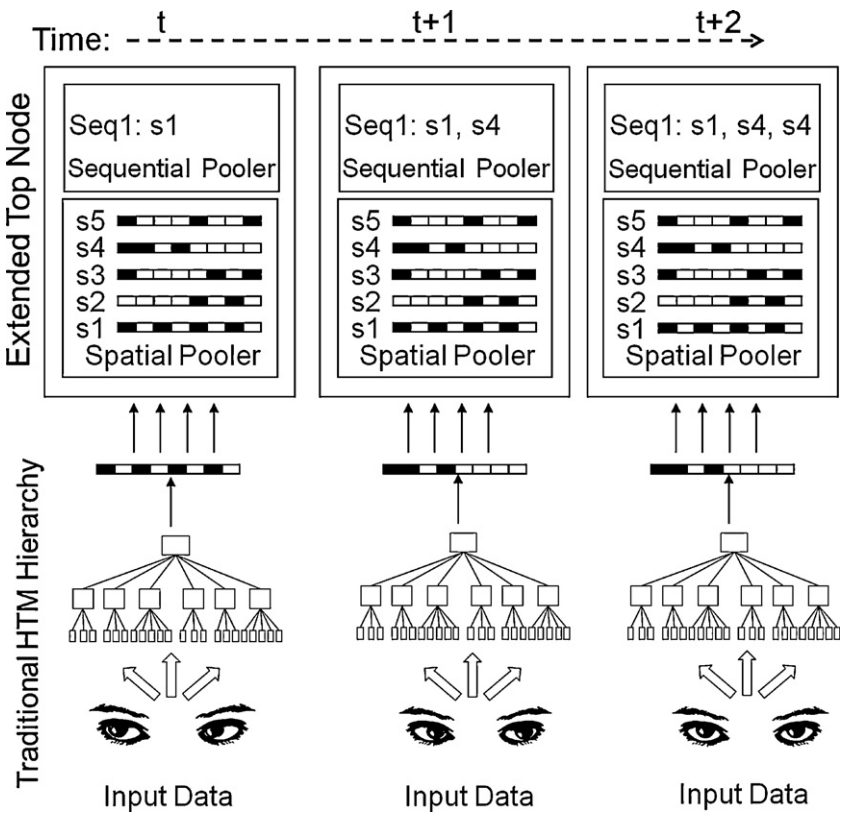

Fig. 5. Extended HTM formalism to capture the temporal structure of data. Traditional top nodes receive binary vectors representing the temporal groups active in the nodes underneath in the hierarchy and map this incoming vectors to categories. The extended top node instead stores sequences of incoming vectors in an abstraction referred to as *sequential pooler* and maps these sequences to categories. The sequences represent the 'utterance' of a sign over time.

the eyes. It is the temporal sequence of PoRs what constitutes a gesture. The different nature of this kind of problem and the sub-optimal performance of traditional HTM networks to deal with it, justified the undertaking of modifications in the HTM inner-workings to adjust the system to the temporal requirements of multi-variable time series and in particular gaze gesture recognition.

The modification of the traditional HTM system concerned the network's top node. In traditional HTM algorithms, the top node just maps incoming vectors from children nodes to categories at any give time. The extended top node instead, stored sequences of incoming vectors from children nodes in an abstraction referred to as '*sequential pooler*' and mapped the sequences as a whole to a category, not just the individual elements. Each element of the sequence is a spatio-temporal configuration pattern of the eyes' PoR at any given instant. The aggregation of this elements form sequences that encapsulate the whole spatio-temporal structure of a gaze gesture as it evolves over time.

Fig. 5 shows an illustration of the abstraction referred to as '*sequential pooler*' in the extended HTM. On it, it can be observed how as time passes, the extended top node threads sequences by incorporating in a linear array the spatio-temporal arrangements of the eyes over time. In this fashion, whole sequences capturing the entire spatio-temporal structure of a gaze gesture from beginning to end are constructed.

Topologically, the extended top node sits at the top of the HTM network, receiving its inputs from a traditional top node underneath serving the purposes of its unique children node, see Fig. 6. As shown in that figure also, traditional HTM require a data structure containing the complete spatio-temporal characterization of a gaze gesture over time. The extended HTM however, receives the data structure of a gaze gesture as it evolves over time.

The preprocessing of the input data for the traditional HTM consisted on transforming the time series of (x, y) gaze coordinates into a spatio-temporal code in a 6×5 2-dimensional matrix. This matrix represents the screen over which the gaze gesture was performed and the elements in the matrix indicate both the temporal and spatial structure of the gaze gesture on the screen. The temporal structure used 3 temporal stages. That consisted of dividing the total time employed during performance of a gesture in 3 slices: beginning (1), middle (2) and end (3) and assigning to the matrix (representing the screen) elements the corresponding numbers (1, 2 or 3) depending when gaze was determined to hovered over a particular area and a 0 if gaze was not determined to hoovered over that area.

Since the spectrum of all possible sequences to store during training and to recognize during inference would quickly overflow the memory available to the node, a means to cluster different sequences into the same category was needed. This clustering was carried out by performing similarity measurements between an incoming sequence and previously stored sequences and classifying the incoming sequence as belonging to a certain cluster (representing a category) of similar sequences according to a threshold. The need for a measurement of similarity was two-fold: It was needed during training in order to determine which sequences to store and which ones to disregard, in case of high similarity to an already stored sequence. A similarity measurement was also needed during inference to determine which sequence, from the set of stored sequences in the sequential pooler of a top node had the highest degree of similarity to an incoming input sequence. The similarity measurements between sequences representing signs were carried out using the Needleman–Wunsch algorithm [33] of dynamic programming for sequence alignment whose inner workings are shown in Fig. 7. Dynamic programming

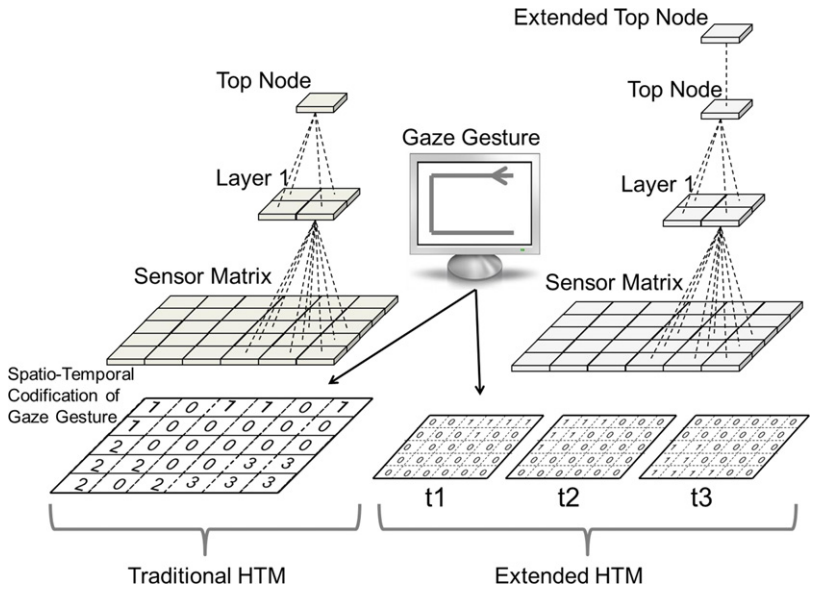

Fig. 6. Extended HTM formalism to capture the temporal structure of gaze gestures data. Traditional top nodes receive binary vectors representing the temporal groups active in the nodes underneath in the hierarchy and map this incoming vectors to categories. The extended top node instead stores sequences of incoming vectors in an abstraction referred to as *sequential pooler* and maps whole sequences to single categories. These sequences represent the "utterance" of a sign over time. Due to the noise presented in the gaze tracker data, the mapping from the intended gaze gesture to the streamed gaze data is not perfect.

has been successfully used by the bioinformatics research community to calculate the degree of similarity between genetic sequences [34]. Dynamic Programming sequence alignment consists of using a scoring matrix to align two sequences according to a scoring scheme by tracing down the optimal global alignment, see Fig. 7.

Fig. 7 shows a sequence alignment calculated with the Needleman–Wunsch algorithm for two sequences of gaze gestures.

$$|D(i,j)| = max \begin{cases} D(i-1,j-1) + s(x_i, y_j) \\ D(i-1,j) + g \\ D(i,j-1) + g \end{cases} \quad (1)$$

The score at the bottom right of the matrix in Fig. 7 indicates the degree of similarity between both sequences for the global alignment. The alignment can be traced back (highlighted in bold in Fig. 7) using a traceback matrix, T, starting from the bottom right element and traveling backwards according to:

$$T(i,j) = argmax \begin{cases} D(i-1,j-1) \\ D(i-1,j) \\ D(i,j-1) \end{cases} \quad (2)$$

3. Experimental setup

During the generation of the gaze gestures data set and during the evaluation of the ability of the recognition algorithms to correctly classify unseen instances, a black or special purpose background was not used, nor markers to attract or better position the gaze in specific coordinates of the screen. The performance of the gaze gesture was always measured over a normal desktop, with

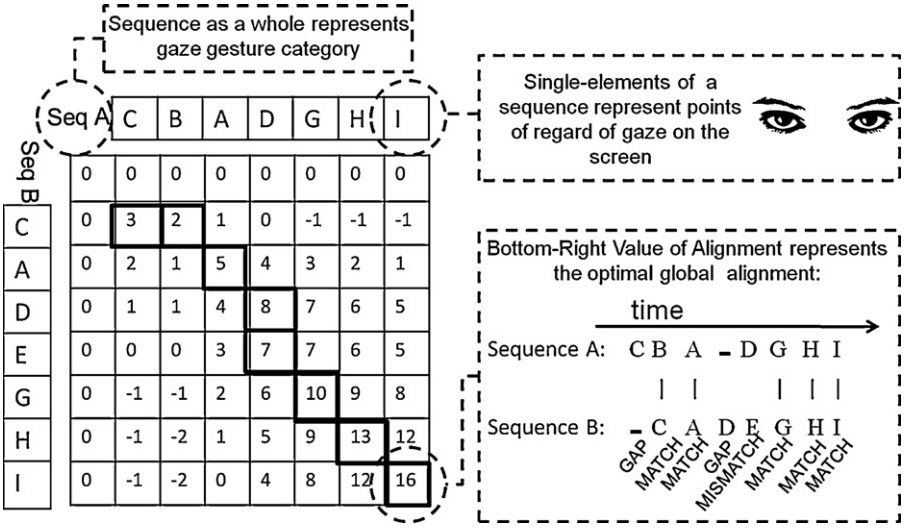

Fig. 7. Sequence alignment with dynamic programming. The extended top node uses dynamic programming for sequence alignment to measure the degree of similarity between two sequences. The external row and column in the matrix represent 2 example sequences encoding the whole spatio-temporal structure of a sign. A system to score matches, mismatches and gaps is used to calculate the sequence alignment matrix. The score at the bottom right of the matrix indicates the top global alignment between both sequences.

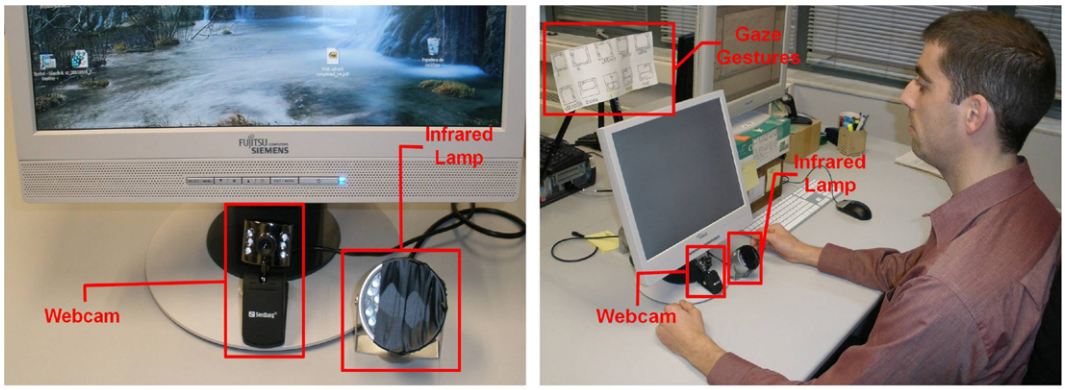

Fig. 8. Experimental setup. Panels a and b show the low cost experimental setup and hardware used to carry out remote eye tracking and the subsequent generation of gaze gestures data. The webcam used has a cost of less than 30$ and simply lacks an infrared filter. The infrared lamp costs less than 5$ and part of it has been covered to decrease the intensity of the illumination.

open windows, panels, programs, etc. This was done to recreate the possible scenario in which potential applications of gaze gestures would take place. It also underlines one of the intrinsic advantages of gaze gestures: they do not take screen real estate.

The remote eye tracking system setup is shown in Fig. 8 and unlike most eye tracking systems, it was a low cost solution of less than 50$. Experiments were carried out using the open source ITU Gaze Tracker [6] in a remote setup to perform gaze tracking. In [29] a detailed video account of the experimental setup and the performance of gaze gestures is presented. The eye image data was captured using an off-the-shelf webcam (Sandberg Nightcam 2), mounted under the computer monitor. One infrared lamp was used to improve pupil-to-iris contrast and to create a glint on the cornea that the gaze tracking algorithm uses as reference to measure pupil center displacements during calibration and tracking. Camera resolution was set to 640×480 pixels, and the frame rate oscillated between 15 and 30 frames per second. In this remote setup, the distance from the eye to the camera was approximately 50–60 cm. The gaze accuracy of the setup was about 1.5°, with marked oscillations among different users. Some users achieved up to 0.7° accuracy while others could not achieve better accuracy than 2°. Theses difference are attributable to individual eye shape, degree of concentrations during calibration, tracking elements arrangement, and light conditions.

The ITU Gaze Tracker streamed all calculated gaze coordinates through a TCP/IP server, see Fig. 1. This raw data was accessed by an in-house developed client that either served the data to the gaze gesture recognition engine or stored the data in text files for off-line analysis and HTM training.

The user study was carried out by 15 participants. 13 of them were male and 2 female, with ages ranging from 20 to 59 years old. All of them used computers regularly, 5 of them were already familiar with eye tracking and 10 of them had never used a gaze tracking system before. All of them were regular computer users. The subset of 10 participants with no previous gaze tracking experience was involved in an experiment about learning effects. All participants had a European or Latin American cultural background and academic education.

During experiments, the performance of two different modalities of gaze gesture was compared: in one, users had to use dwell time at the beginning and at the end of the gesture to signal the segmentation of the gesture within other types of gaze activity; This requirement obviously makes recognition easier for the algorithm but increases the load on the user and the time required to carry out the gesture. In the alternative modality, users had to perform the gaze gesture without using dwell time to indicate beginning and end of a gesture, making it more comfortable and faster to carry out the gesture by the user, but making the recognition more challenging. The dwell time threshold was set to 400 ms. In the dwell time modality, the user was notified by audio feedback when the dwell time threshold was surpassed. In summary, two different experimental conditions were studied in the experiment: *saccadic gaze gestures without dwell time*, and *saccadic gaze gestures with dwell time*.

The performance of traditional and extended HTMs was determined by measuring after training each network, what percentage of unseen signs' instances from the experimental subjects were assigned to their proper categories using the inference scores provided by the corresponding algorithm on inference mode.

Participants were asked to complete 3 different tasks, an *accuracy task*, a *velocity task* and a *browsing task*. In the accuracy task, users had to perform the sequence of 10 gaze gestures shown in Fig. 9 for each experimental condition. Participants were instructed to complete the gestures as fast and as accurately as possible. Users were notified with audio feedback every time the system detected a gesture, regardless of whether the detected gesture was correct or not. When a gesture was detected, the user had to proceed to perform the next gesture. For the accuracy task, the *accuracy* was determined as the percentage of correct gestures detected.

In the velocity task, participants were instructed to perform the same gesture 10 times as fast and as accurately as possible. Three representative gestures were selected for this task: 'c', 'j,' and 'hook'. These gestures were chosen for representing gestures with 3, 4 and 5 strokes, see Fig. 9. In each velocity trial, the approximate *time per gesture*, or *TPG*, measured in seconds was calculated by dividing by 10 the total time taken to perform 10 repetitions of the same gesture.

The browsing task required participants to browse the Internet during 5 min. During this time the number of *involuntary* gaze gestures detected, i.e. false positives, was measured for each of the 2 experimental conditions.

A subset of 10 participants with no previous eye tracking experience, repeated the accuracy and velocity task over 5 blocks to study learning effects.

4. Results

The data set of gaze gestures to train the HTM networks was generated by 1 user who performed 50 instances of each category. The aggregated sum of all instances for each category is shown in

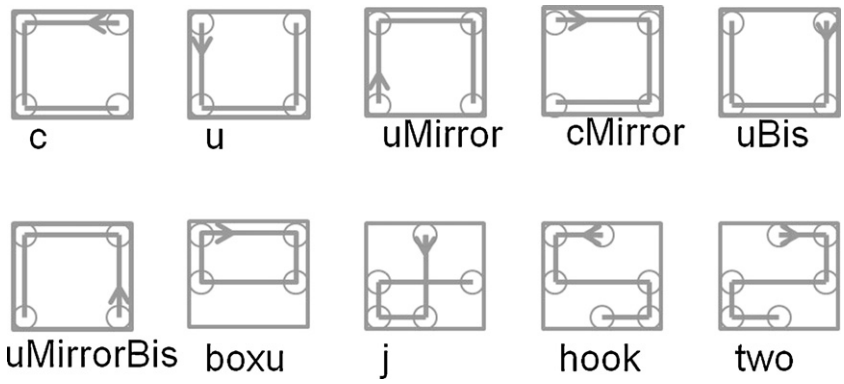

Fig. 9. Gestures set. Set of 10 gaze gestures employed in the user study to evaluate the performance of different gaze gestures modalities. Circles indicate places were users perform brief fixations. This specific set of gestures were selected for representing an intermediate point between simple and complex gaze gestures.

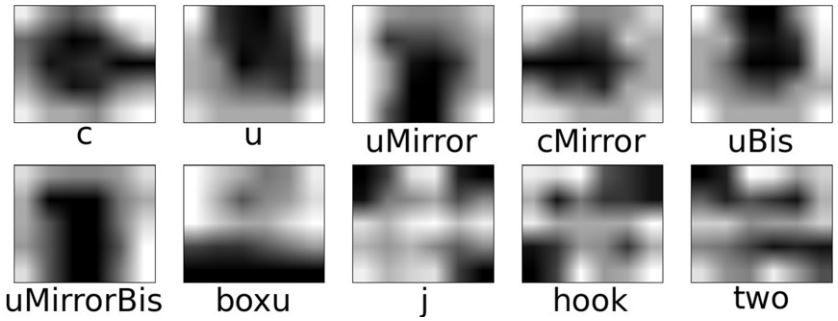

Fig. 10. Gaze gestures instances. Visualization of the degree of overlap of the 50 instances per category of gaze gestures data used for training the HTM networks. A certain degree of noise in clearly visible in the data.

Fig. 10. The lack of a perfect representation of the idealized gestures (see Fig. 9 in Fig. 10 illustrates the limitations in terms of resolution and accuracy of gaze tracking technology and the presence of noise in the data. This constitutes and additional challenge for the recognition algorithm.

HTM theory predicts the ability of HTM networks to warp time by increasing the number of layers in the topology [35]. That is, a higher number of layers in the topology should extend the temporal invariance of the algorithm at the higher nodes of the network and hence, the HTM algorithm should perform learning and inference on problems where data structures slowly unfold over time. To test this theoretical prediction, a number of simulations testing the recognition accuracy of traditional HTM networks with increasing number of layers in their topology were carried out. As can be seen in Fig. 11, increasing the number of layers appears to improve performance on the problem at hand for traditional HTMs up to the 2 Layers level, with a statistically significant effect of $F(5, 84) = 9.33$, $p < 0.05$. The fact that a two layer network shows the most optimal performance is good in terms of maintaining to a minimum the computational complexity of the algorithm. Still, the recognition accuracy of traditional HTM even at the optimal 2 layers level is not satisfactory. This first experiment highlighted the necessity for HTM algorithms to be modified in order for them to better accommodate recognition problems where instances unfold over time. Further experiments were all carried out with 2 Layer network topologies as the ones shown in Fig. 6. Recognition was carried out with traditional HTM and with the extended HTM.

Fig. 12 shows the results obtained for the accuracy task. The extended HTMs with no dwell time had an average accuracy of 94% and the extended HTMs with dwell time had an average accuracy of 98%. As can be seen in Fig. 12, the extended HTM algorithm clearly outperforms the traditional HTM in terms of accuracy, $F(3, 56) = 39.42$, $p < 0.05$.

To measure the time needed to complete a gesture, we used the measurement: time per gesture (TPG). Three representative gestures with 3, 4, and 5 strokes were selected from the data set from Fig. 9, namely 'c', 'j' and 'hook'. The results are shown in Fig. 13. Average TPG for gestures 'c', 'j' and 'hook' with no dwell time were 0.8, 1.3 and 1.9 s respectively. Using dwell times to signal beginning and end of gestures generated TPGs of 2, 2.4 and 3.1 s respectively. Gesture modality (with or without dwell) had a significant effect on TPG for 'c', 'j' and 'hook' with $F(1, 28) = 274$, 289 and 265 respectively, $p < 0.05$. Gesture type also had a significant effect on

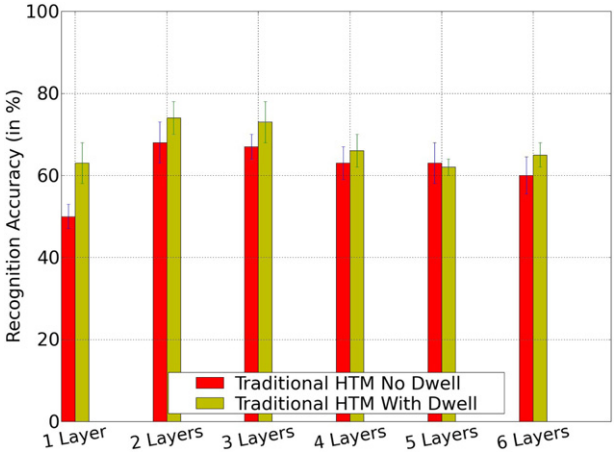

Fig. 11. Optimal topology search. Gaze gestures recognition accuracy for traditional HTM networks with different number of layers for the experimental conditions with and without dwell. The bars show the percentage of correct classifications achieved by 1, 2, 3, 4, 5 and 6 Layer networks. As can be seen in the graph, best performance for traditional HTMs is achieved by topologies with 2 Layers.

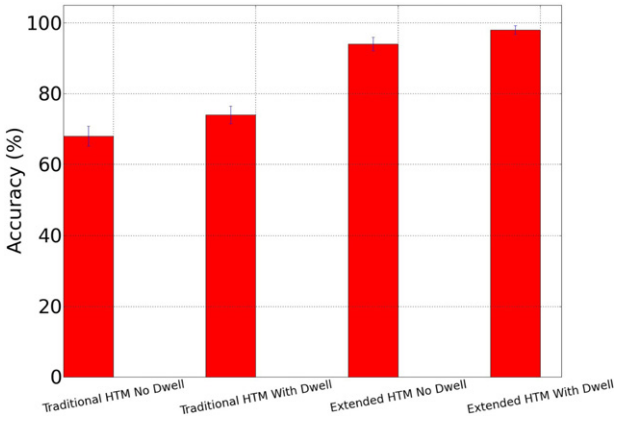

Fig. 12. Accuracy experiment. Average recognition accuracy for each of the four conditions in the experiment. Error bars show the standard error of the mean.

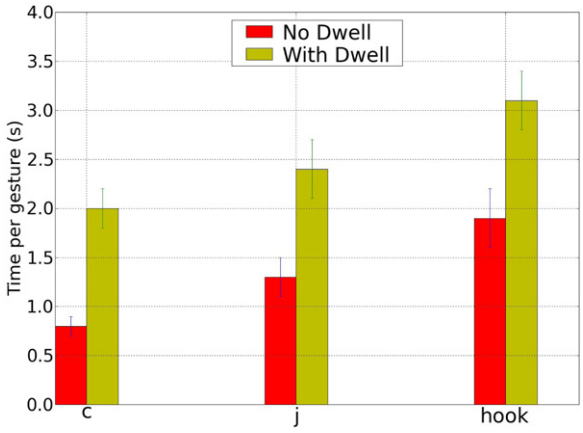

Fig. 13. Velocity experiment. Average TPG for 3, 4 and 5 stroke gestures recognized with the Extended HTM algorithm. Error bars show the standard error of the mean. Note that the modality with dwell includes the time needed for the two long fixations to signal beginning and end of a gesture (400 ms each).

TPG, $F(1, 28) = 274$, $p < 0.05$, with shorter gestures being faster to complete.

Fig. 14 shows the results for the browsing experiment, plotting the number of involuntary gaze gestures per minute for the extended and traditional HTMs with and without dwell during 5 min of Internet browsing. There was a significant effect of gesture

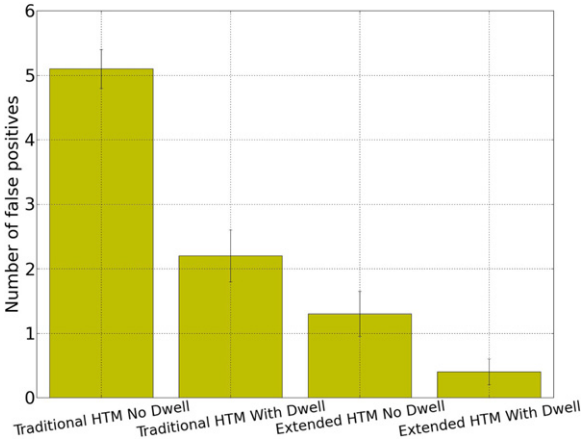

Fig. 14. Browsing experiment. Average number of involuntary gaze gestures detected during 5 min of Internet browsing for each of the experimental conditions. Error bars show the standard error of the mean.

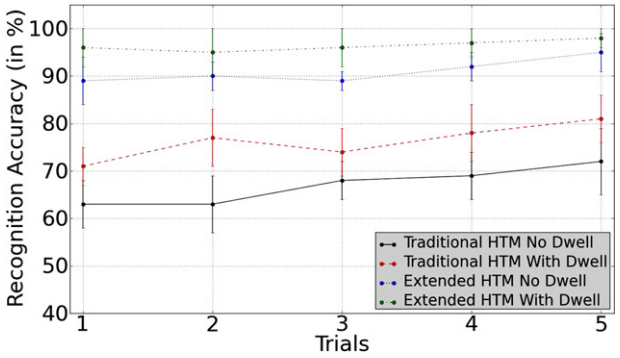

Fig. 15. Learning curve on accuracy. Average recognition accuracy for each experimental condition over 5 blocks for 10 users with no previous gaze tracking experience.

technique (with or without dwell) in the number of involuntary gestures detected for both traditional HTM and extended HTM, $F(1, 28) = 69.14$ and 16.73 respectively, $p < 0.05$. There was also a statistically significant overall effect $F(3, 56) = 95.01$, $p < 0.05$. The best performance (less false positives detected) was obtained by the gesture modality using dwell time and recognized with the extended HTM.

A sub-study with 10 participants with no previous eye tracking experience was carried out to study learning effects. Learning effects were studied for both accuracy and TPG. There was no clear learning effect in terms of accuracy over the 5 experimental blocks (Fig. 15), but there is a noticeable, albeit slight, learning effect in terms of TPG over the 5 experimental blocks (Fig. 16).

The impact of vocabulary size on recognition accuracy was also studied, i.e. how increasing the number of categories to be recognized negatively affects performance for the different algorithms being compared. As can be seen in Fig. 17 the effect of degrading performance is more noticeable for traditional HTMs with extended HTMs performing more robustly.

HTM requires training sets where each category to be classified is represented by several instances. To determine the effect on recognition accuracy of increasing the number of training instances for each method being compared, the results of simulations in which the number of training instances was increased in discrete steps were plotted. As can be observed in Fig. 18, increasing the number of training instances improves performance in general but the extended HTM consistently outperforms the traditional HTM methodology.

An experiment was designed in which 150 instances from voluntarily performed gaze gestures and 150 instances of normal gaze activity were generated. The recognition scores generated by the traditional and extended HTM algorithm during inference under

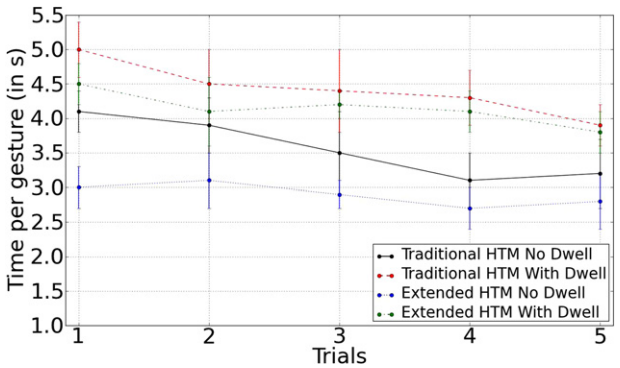

Fig. 16. Learning curve on TPG. Average TPG for each experimental condition over 5 blocks for 10 users with no previous gaze tracking experience.

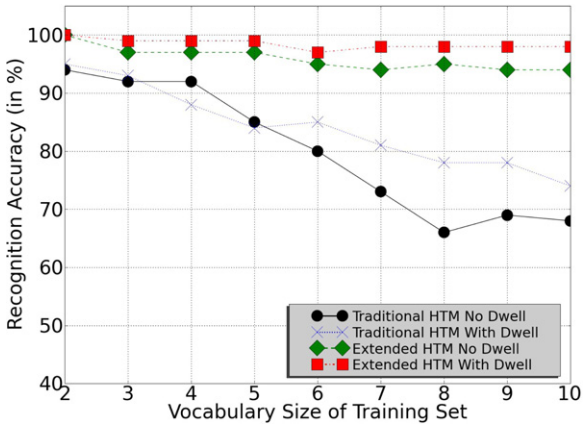

Fig. 17. Decreasing performance with increasing vocabulary size. The recognition accuracy for each method being compared decreases as the vocabulary size of the sign gestures data set to be learnt increases. Yet, this effect is more noticeable for Traditional HTMs.

the experimental condition of saccadic gaze gestures with no dwell time were plotted in a histogram. Fig. 19 shows how the range of scores obtained by normal gaze activity overlaps over a significant sub-range with the range of scores obtained by intentional gaze gestures when using the traditional HTM as recognition algorithm.

Fig. 20 shows the range of scores obtained for saccadic gaze gestures and standard gaze activity using the extended HTM algorithm. As can be observed, there is a clear separation between scores obtained for intended gaze gestures and the rest of gaze activity. This means, that with the extended HTMs a clear cutoff threshold for inference values exists that is able to discriminate most intentional gaze gestures from non-gaze gestures activity.

5. Discussion

The low cost hardware employed in the experiments generated noisy gaze gestures data that demanded a robust and noise-tolerant recognition engine. Also, the spatio-temporal characteristics of gaze gestures require a recognition engine that is able to learn classes with a temporal structure. In this work, an alternative approach to how HTMs handle data at the top node has been presented. This extension was developed in order to improve the performance of HTMs on machine learning tasks in which instances are composed of a noisy sequence of patterns that unfold slowly over time.

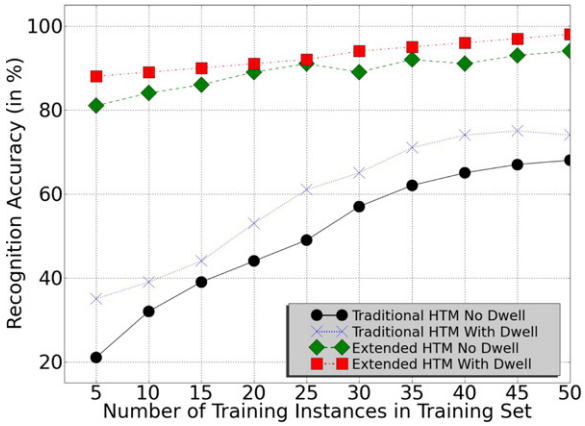

Fig. 18. Improving performance with increasing number of training instances. The recognition accuracy for each method being compared increases as the number of training instances gets larger.

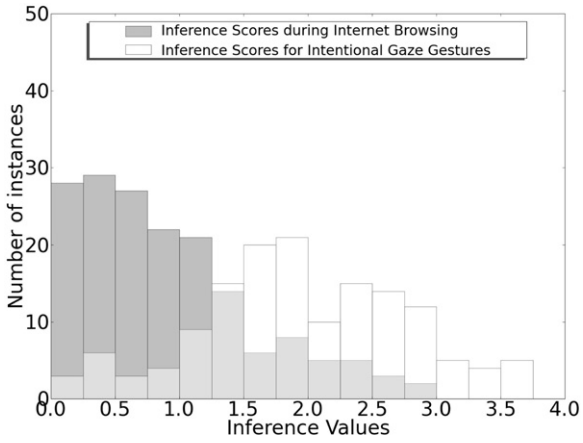

Fig. 19. Traditional HTM recognition scores distribution for gaze gestures and standard gaze activity while browsing the Internet. The dark grey bars correspond to the histogram distribution of inference values generated by traditional HTM networks when fed with normal gaze activity generated during standard user–computer interaction. The white bars correspond to the distribution of inference values generated by the HTM networks when fed with a wide array of consciously performed gaze gestures. Light grey areas correspond to a region of overlap between both distributions.

We acknowledge that other paradigms exists that perform sequence classification: Hidden Markov Models (HMM) [36,37] or recurrent neural networks (RNN) [20] such as Elman networks or Jordan Networks. The disadvantages of HMMs are that the types of prior distributions that can be placed on hidden states are severely limited and often it is not possible to predict the probability of seeing an arbitrary observation. Local optima is a more significant problem in RNN than in feed-forward neural networks [38].

HTM contains several key bioinspired insights that are incorporated into the model and differentiate the extended HTM from canonical RNNs such as the usage of a hierarchical structure, the observance of the time dimension to automatically cluster temporally adjacent spatial patterns and in their current version, that just uses binary representations to codify data, a very light weight in terms of computational resources needed. In more advanced HTMs, such as those proposed in [22], the usage of sparse representations constitutes and additional bioinspired property used by the algorithm. Furthermore, the extended HTM can compare favorably with

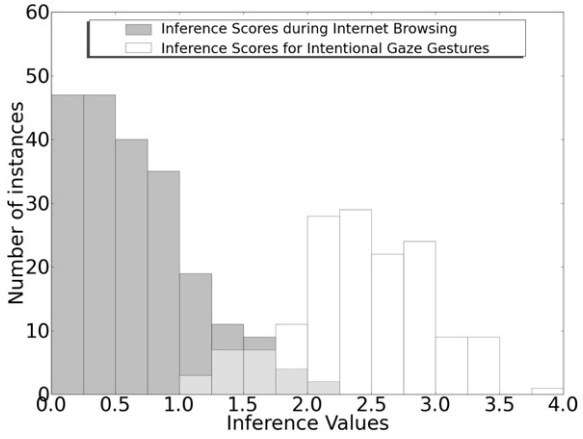

Fig. 20. Extended HTM recognition scores distribution for gaze gestures and standard gaze activity while browsing the Internet. The dark grey bars correspond to the distribution of inference values generated by the extended HTM fed with normal gaze activity generated during standard user–computer interaction. The white bars correspond to the distribution of inference values generated by the extended HTM algorithm when fed with consciously performed gaze gestures. The light grey areas correspond to a minimal region of overlap between both distributions.

HMMs in learning temporal sequences [39], showing better performance with less training instances, and being more robust to increasing vocabulary sizes [28]. HTM networks tend to be compact in size, with few nodes sampling the input space, have few parameters to tune in [26] in comparison to other ANNs, and are often light weight during inference in terms of offering real time efficiency.

The extension of a traditional HTM system proposed here consists of a reformulation of the network's top node that allows it to store sequences of input patterns constituting whole instances of the data set and to map these sequences to categories from the category space. Instances from the whole input space are stored and compared using Dynamic Programming that allows for dynamic time warping. Doing this, the top node is able to warp time and create temporal invariance with robust recognition of instances with a temporal structure and performed at different velocities.

In gaze gesture recognition, it is the orderly sequence of gaze vectors arriving over time what constitutes a gesture, as oppose for instance to image recognition, where at any time point, the input vector represents a complete characterization of an image category. As can be seen in Fig. 12, the extended HTM algorithm clearly improves the performance of traditional HTMs for gaze gesture recognition. Although in this work the focus has been on input data coming from an eye tracker, HTM theory in general and the extended HTM approach in particular are very flexible in terms of accepting input data streams from a variety of sensors [28]. As long as the data representations in the input contains a spatio-temporal structure, the extended HTM approach, after training with a sufficient number of training instances, can achieve good recognition accuracy (e.g. up to 98% as shown by experimental results). The method is also light on computing resources during inference, requiring only milliseconds, which makes it easily applicable to real-time requirements contexts.

Recognition of isolated gaze gestures is a relatively trivial task for both traditional HTMs and the extended HTM system [40]. The challenging problem is gaze gesture recognition in real time. That is, to distinguish intentional gaze gestures intertwined with typical gaze activity during normal gaze interaction with a computer. This requires finding the right trade-off between sensitivity and specificity of the recognition algorithms. On the one side, the aim is to detect the maximum amount of intentionally performed gestures, that is, to maximize sensitivity. On the other hand, it is essential to minimize false positives, that is, to maximize specificity. This challenge was only partially overcome by the traditional HTM during experiments, but it was robustly overcome by the extended HTM as shown by the accuracy, velocity and browsing tasks results in Figs. 12–14.

The suboptimal performance of traditional HTM recognition of gaze gestures was due to several factors. First, the noisy nature of the raw data generated by the eye tracker and the low resolution of the data structure representation matrix, 6×5, created a significant degree of overlap between several instances from different categories of gaze gestures, see Fig. 10. This constitutes a challenge for any type of recognition algorithm. A naive solution would be to increase the granularity of the data structure. However, this would require gathering more training data from users since the degree of overlap among different instances within a category with increased granularity would decrease, and therefore, clustering same category instances would be harder for the HTM algorithm. Due to limited resources availability to gather additional training data, experiments were constrained to 50 instances per category of gaze gestures and the 6×5 matrix data structure.

Decreasing the detection threshold of the similarity score produced by the recognition algorithm increases the chances of detecting a gesture when it is performed but also increases the amount of false positives generated by the algorithm. On the other hand, increasing the detection threshold of this similarity score produced by the recognition algorithm increases the strictness of the similarity, lowering the number of false positives produced, but also missing some true positives that are noisy or not accurate enough and hence only obtain a low similarity score. An obvious solution to circumvent this problem is to impose on the user the need to indicate through an external switching action, the beginning and end of a gaze gesture. This adds a load on the user and increases the time needed to perform a gesture but also simplifies the task of recognition for the algorithm, which obtains the segmentation values of where a gaze gesture starts and ends. This solution is not always appropriate since the amount of switching channels available to persons with disabilities is markedly limited and in some cases, such as locked-in patients, non existent. A switch can be simulated by a fixation detection algorithm and dwell time. In this way, the user indicates the beginning and the end of a gaze gesture through dwell activation at the beginning and end of a conscious gaze gesture. As Figs. 12 and 13 show, this strategy improves accuracy performance significantly but also increases time per gesture, TPG.

The suboptimal performance of traditional HTM algorithms on real time gaze gesture recognition is mainly due to the degree of overlap between the range of inference scores generated by the algorithm for the non-gesture gaze activity and the range of inference scores generated by the algorithm for intentional gaze gestures, as illustrated by Fig. 19. For the offline recognition of gaze gestures, the value of the inference score [35] produced by the HTM algorithms for the top guess is not critical, as long as it is the largest of the list of category guesses that the algorithm generates. During inference on different instances of consciously performed gaze gestures, a wide dispersion of similarity scores emerges, see Fig. 19. That range partially overlaps with inference values from normal gaze activity. Hence, it becomes impossible to determine a threshold able to discriminate gaze activity unrelated to gestures with an intentional gaze gesture for the range of scores generated by the traditional HTM.

The good performance of the extended HTM algorithm was due to the inherent ability of dynamic programming algorithms in the top node to warp time and become invariant to sequences that can be performed over different time scales. This translated on a clear distinction of alignment scores obtained by conscious gaze gestures and the rest of gaze activity, as shown in Fig. 20. The clear partition between both distributions results in a straight forward way to discriminate normal gaze activity from consciously performed gaze gestures by choosing a threshold in the middle of the overlapping range of scores as a way to minimize false positives and maximize the sensitivity of gaze gesture detection. Additional advantages of the extended HTM over traditional HTM is the recognition accuracy robustness of the former to increasing data set vocabularies, i.e. number of recognition categories, see Fig. 17, and lesser requirements in terms of number of training instances in the training set, see Fig. 18.

Even though the extended HTMs achieved better recognition accuracy for gaze gestures than traditional HTMs, the inference time of the extended HTM algorithm was worse than traditional HTMs. This is due to the demanding nature of the Needleman–Wunsch algorithm [33] for sequence alignment employed by the extended top node. Nonetheless, due to the very small sequences (5–7 elements long) and small data sets (10 categories), employed in the experiments, computing power was not a limiting factor and recognition time during inference was still in the order of milliseconds. This fast inference capability of the extended HTM algorithm is important for real time applications.

Human beings in general are not used to employ gaze as an output actor; rather, gaze is normally employed for pointing and targeting and then as an input sensor. However, with a bit of

training an adaptation, this issue can be easily overcome in a couple of sessions to the extent of gaze gestures becoming a relatively straight forward way of emitting commands. The performance improvement in terms of TPG after just 5 short (less than 10 min each) experimental sessions is clearly visible in Fig. 16 for gaze gestures with and without dwell time and illustrates the fast assimilation of gaze gestures as a form of output communication.

Even though the saccadic gaze gestures with no dwell time generated some false positives (Fig. 14) users reported to prefer not to use fixations because of the faster nature and less stressful nature of not using dwell-activated fixations to signal the start and end of a gesture. The dwell threshold time used during experiments was fixed. In a real scenario, users should be able to set their preferred dwell time to signal the beginning and end of a gesture. In this case, users might trade off a decrease in speed for a lower rate of false positives. The proper approach should in any case find a balance between the costs of making an error with the likelihood of making one [5].

At the current state of the technology, the use of gaze gestures is constrained to a limited group of users or environments. For normal users that can employ their hands in standard environments, traditional devices to generate commands such as keyboard and mouse offer superior performance in terms of robustness and intuitiveness. Nonetheless, as mobile electronic devices become more pervasive and powerful, their inherent reduced display size makes them obvious candidates for gaze gestures as an alternative option to generate control commands. Moreover, in certain environments where use of the hands may be limited, such as a surgery room or living rooms equipped with media centers, eye tracking technology and gaze gesture recognition as a sub-application of the field might offer considerable value.

For users with severe disabilities, such as those with high vertebrae spinal cord injuries ALS, brain-stem stroke and particularly locked-in patients whose gaze constitutes the only output communication channel available, gaze gestures provide a much needed output modality. Gaze gestures can provide this group of users with increasing autonomy in the areas of communication, environmental control and mobility. Video-based eye tracking technology has so far allowed these users to position a cursor on the screen with up to 0.5° accuracy [8]. However, the emission of switching actions are challenging for this group of users. Hence, using gaze gestures to generate commands, such as 'Enter', 'Next Page', 'Escape' or even whole macros/scripts, constitutes a valuable addition to the limited set of information channel available for these users. Moreover, using gaze gestures as a communication channel between users and computers offers several advantages in terms of speeding up command completion, freeing up on-screen real-estate and avoiding the stressful Midas Touch problem inherent to dwell selection in gaze interaction. Additionally, the hardware/software approach used in this work has used extremely low cost hardware and software components, making it easily adoptable for all types of users regardless of financial condition.

6. Conclusion

In this work an extension of the traditional HTM paradigm has been proposed in order to improve HTM's performance in the real time recognition of saccadic gaze gestures tracked with a low cost and noisy eye tracking system. The two dimensions used to compare the traditional and extended HTM methods, recognition accuracy of the algorithm and false positives rate, show that the extended HTM method outperforms significantly traditional HTMs by being more accurate and less error prone in preventing accidental algorithmic recognition of unintentional gaze gestures within the context of real time gaze computer interaction.

The motivation for an extended HTM was two fold: to adapt HTM algorithms to problems were the spatio-temporal structure of instances unfolds over time and to build a system able to robustly recognize gaze gestures in real time. The purpose of such system is to provide a communication channel between humans and computers, targeting specifically either users with disabilities or environments where traditional input channels are not suitable or could be augmented.

Severely paralyzed individuals unable to use a switch but still able to use their gaze as a mono-modal input to control a computer could benefit from gaze gestures as an additional interaction modality. In addition, situations in which calibration is not possible, such as interactive displays for random users walking by, represent another potential setup where gaze gestures could be of use. Moreover, gaze gestures could be incorporated in the context of multi-modal input as an additional interaction modality for mobile devices with limited screen real estate such as smartphones or tablet computers.

Gaze gestures are not intrinsically intuitive since humans rarely use their eyes for conscious effector purposes beyond gaze pointing. Yet, the experimental results shown here prove that users quickly become familiar with a medium size set of gaze gestures. An advantage of using gaze gestures for HCI is that gaze gestures are not affected by the low accuracy problems intrinsic to video-based eye tracking systems. Gaze gesture recognition does not require high tracking accuracy and can even be implemented in the absence of a calibration procedure. In addition, the 6 muscles involved in eye movements are know to be particularly resistant to strain and degenerative conditions [5,15], hence constituting a reliable method for HCI. Furthermore, gaze gestures do not occupy real estate on the screen, freeing up therefore space for other purposes. Hence, gaze gestures constitute an innovative modality of HCI that can reliably convey information from the eyes to a computer using the extended HTM algorithmic for recognition purposes.

Last but not least, the extended HTM approach is not specifically designed to deal with gaze gestures specifically, since it is highly independent of hardware and preprocessing of input data. In fact, extended HTMs possess a high degree of flexibility and can be easily adapted to a wide array of machine learning applications where the patterns to be learned are multi-variable time-series unfolding slowly over time.

Acknowledgements

This work was supported by grants from 'Consejería de Educación de la Comunidad de Madrid' (C.A.M), the 'European Social Fund (E.S.F.)' and the 'Ministerio de Ciencia e Innovacion': MICINN BFU2009-08473 and TIN 2010-19607.

References

[1] H. Drewes, A. Schmidt, Interacting with the computer using gaze gestures, in: Proceedings of the 11th IFIP TC 13 International Conference on Human–Computer Interaction – Volume Part II, INTERACT'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 475–488.
[2] T. Kapuscinski, M. Wysocki, Using hierarchical temporal memory for recognition of signed polish words, in: M. Kurzynski, M. Wozniak (Eds.), Computer Recognition Systems: Advances in Intelligent and Soft Computing, Springer-Verlag, Berlin/Heidelberg, 2009, pp. 355–362.
[3] W. Zheng, Z.-z. Cui, Z. Zheng, Y. Zhang, Remote monitoring of human hand motion using induced electrostatic signals, Journal of Electrostatics 69 (6) (2011) 571–577.
[4] N.H. Dardas, N.D. Georganas, Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques, IEEE Transactions on Instrumentation and Measurement 60 (11) (2011) 3592–3607.
[5] E. Mollenbach, Selection Strategies in Gaze Interaction, PhD Thesis, Loughborough University, 2010.
[6] J. San Agustin, H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D.W. Hansen, J.P. Hansen, Evaluation of a low-cost open-source gaze tracker, in: ETRA '10:

Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ACM, New York, NY, USA, 2010, pp. 77–80.
[7] A.T. Duchowski, Eye Tracking Methodology: Theory and Practice, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
[8] C. Hennessey, B. Noureddin, P. Lawrence, Fixation precision in high-speed noncontact eye-gaze tracking, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38 (2) (2008) 289–298.
[9] R.J.K. Jacob, The use of eye movements in human–computer interaction techniques: what you look at is what you get, ACM Transactions on Information Systems 9 (2) (1991) 152–169.
[10] E. Mollenbach, J.P. Hansen, M. Lillholm, A.G. Gale, Single stroke gaze gestures, in: Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems, CHI '09, ACM, New York, NY, USA, 2009, pp. 4555–4560.
[11] P. Qvarfordt, S. Zhai, Conversing with the user based on eye-gaze patterns, in: CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, 2005, pp. 221–230.
[12] J.O. Wobbrock, J. Rubinstein, M. Sawyer, A.T. Duchowski, L. Uk, Gaze-based creativity not typing but writing: eye-based text entry using letter-like gestures, in: The 3rd Conference on Communication by Gaze Interaction – COGAIN 2007, 2007.
[13] N. Bee, E. André, Writing with your eye: a dwell time free writing system adapted to the nature of human eye gaze, in: Proceedings of the 4th IEEE Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems, Springer-Verlag, 2008, pp. 111–122.
[14] P. Isokoski, Text input methods for eye trackers using off-screen targets, in: ETRA '00: Proceedings of the 2000 Symposium on Eye Tracking Research & Applications, ACM, New York, NY, USA, 2000, pp. 15–21.
[15] H. Drewes, Eye Gaze Tracking for Human Computer Interaction, PhD Thesis, Ludwig-Maximilians-Universität München, 2010.
[16] J.O. Wobbrock, J. Rubinstein, M.W. Sawyer, A.T. Duchowski, Longitudinal evaluation of discrete consecutive gaze gestures for text entry, in: Proceedings of the 2008 Symposium on Eye Tracking Research & Applications, ACM, Savannah, Georgia, 2008, pp. 11–18.
[17] S. Vickers, H. Istance, A. Hyrskykari, L. Immonen, S. Mansikkamaa, Designing gaze gestures for gaming: an investigation of performance, in: Proceedings of the 2010 Symposium on Eye Tracking Research & Applications; ETRA 2010, ACM Press, Austin, TX, 2010.
[18] B. Srinath Reddy, O.A. Basir, Concept-based evidential reasoning for multimodal fusion in human–computer interaction, Applied Soft Computing 10 (2) (2010) 567–577.
[19] G. Fang, W. Gao, A srn/hmm system for signer-independent continuous sign language recognition, in: Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 2002. Proceedings. May, 2002, pp. 312–317.
[20] T.-J. Hsieh, H.-F. Hsiao, W.-C. Yeh, Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm, Applied Soft Computing 11 (March (2)) (2011) 2510–2525.
[21] R.-H. Liang, M. Ouhyoung, A real-time continuous gesture recognition system for sign language., in: Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998. Proceedings. April, 1998, pp. 558–567.
[22] D. George, J. Hawkins, Towards a mathematical theory of cortical micro-circuits, PLoS Computational Biology 5 (10) (2009) e1000532, 10.
[23] J. Hawkins, Hierarchical temporal memory, concepts, theory, and terminology, Technical report, Numenta, 2006.
[24] M.I. Rabinovich, P. Varona, A.I. Selverston, H.D.I. Abarbanel, Dynamical principles in neuroscience, Reviews of Modern Physics 78 (4) (2006).
[25] F.B. Rodríguez, R. Huerta, Analysis of perfect mappings of the stimuli through neural temporal sequences, Neural Networks 17 (7) (2004) 963–973.
[26] D. George, J. Hawkins, A hierarchical Bayesian model of invariant pattern recognition in the visual cortex, in: 2005 IEEE International Joint Conference on Neural Networks, 2005. IJCNN '05. Proceedings, vol. 3, July–4 August, 2005, pp. 1812–1817.
[27] Numenta, Problems that fit htm. Technical report, Numenta, 2006.
[28] D. Rozado, F.B. Rodriguez, P. Varona, Extending the bioinspired hierarchical temporal memory paradigm for sign language recognition, Neurocomputing 79 (0) (2012) 75–86.
[29] D. Rozado, Remote gaze gestures. http://www.youtube.com/watch?v=BaZx2aKoxDI, November 2011.
[30] J. San Agustin, Off-the-Shelf Gaze Interaction, PhD Thesis, IT University of Copenhagen, 2010.
[31] E. Mollenbach, M. Lillholm, A. Gail, J.P. Hansen, Single gaze gestures, in: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA '10, 2010, pp. 177–180.
[32] D. George, J. Hawkins, Belief propagation and wiring length optimization as organizing principles for cortical microcircuits, Technical report, Numenta, http://www.numenta.com, 2006.
[33] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology 48 (3) (1970) 443–453.
[34] R. Giegerich, A systematic approach to dynamic programming in bioinformatics, Bioinformatics 16 (8) (2000) 665–677.
[35] D. George, B. Jarosy, The HTM learning algorithms, Technical report, Numenta, 2007.
[36] M. AL-Rousan, K. Assaleh, A. Tala'a, Video-based signer-independent arabic sign language recognition using hidden markov models, Applied Soft Computing 9 (3) (2009) 990–999.
[37] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, J. Yang, A framework for hand gesture recognition based on accelerometer and EMG sensors, IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans 41 (6) (2011) 1064–1076.
[38] M.P. Cuéllar, M. Delgado, M.C. Pegalajar, An application of non-linear programming to train recurrent neural networks in time series prediction problems, in: C.-S. Chen, J. Filipe, I. Seruca, J. Cordeiro (Eds.), Enterprise Information Systems VII, Springer, Netherlands, 2006, pp. 95–102.
[39] D. Rozado, F. Rodriguez, P. Varona, Optimizing hierarchical temporal memory for multivariable time series, in: K. Diamantaras, W. Duch, L. Iliadis (Eds.), Artificial Neural Networks – ICANN 2010, volume 6353 of Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, 2010, pp. 506–518. doi:10.1007/978-3-642-15822-3-62.
[40] D. Rozado, F. Rodriguez, P. Varona, Gaze gesture recognition with hierarchical temporal memory networks, in: International Work Conference on Artificial Neural Networks 2011 – Lecture Notes in Computer Science, 2011.