

# Comparison of Attention Behaviour Across User Sets through Automatic Identification of Common Areas of Interest

Prithiviraj K. Muthumanickam, Jouni Helske, Aida Nordman, Jimmy Johansson, Matthew Cooper  
University of Linköping, Sweden

[prithiviraj.muthumanickam@liu.se](mailto:prithiviraj.muthumanickam@liu.se), [aida.vitoria@liu.se](mailto:aida.vitoria@liu.se), [jimmy.johansson@liu.se](mailto:jimmy.johansson@liu.se), [matthew.cooper@liu.se](mailto:matthew.cooper@liu.se), [jouni.helske@liu.se](mailto:jouni.helske@liu.se)

## Abstract

*Eye tracking is used to analyze and compare user behaviour within numerous domains, but long duration eye tracking experiments across multiple users generate millions of eye gaze samples, making the data analysis process complex. Usually the samples are labelled into Areas of Interest (AoI) or Objects of Interest (OoI), where the AoI approach aims to understand how a user monitors different regions of a scene while OoI identification uncovers distinct objects in the scene that attract user attention. Using scalable clustering and cluster merging techniques that require minimal user input, we label AoIs across multiple users in long duration eye tracking experiments. Using the common AoI labels then allows direct comparison of the users as well as the use of such methods as Hidden Markov Models and Sequence mining to uncover common and distinct behaviour between the users which, until now, has been prohibitively difficult to achieve.*

## 1. Introduction

Eye-tracking has become a valuable tool to analyze and compare visual scanning behaviour across multiple users in numerous application domains including user interface evaluation studies, air traffic control, driving and maritime simulators etc. Typically, the process involves manual specification of regions of interest in the displayed application by an analyst and the eye-tracking system then recording how the user's visual attention moves between these pre-defined regions as the session progresses. This can be an effective approach but misses a level of detail which goes beyond the analyst's expectations of user behaviour. This has been a focus of recent research which has looked at ways to derive the areas of interest (AoIs) directly from the data, rather than through the analyst's expectations [1, 2]. This approach can, therefore, better capture the user's attention over time. Earlier methods [2] identified the AoIs in long duration eye tracking experiments using

scalable clustering and cluster merging. The results have been demonstrated across gaze data collected from Air Traffic Controllers working in a simulated remote tower environment [3]. But manual comparison of visual scanning behaviour across multiple air traffic controllers was a complex process due to the fact that the IDs or labels of the AoIs found for a single subject do not correlate with those of others.

In this work, we have developed a method for the cross-correlation of AoIs between sets of users performing long-duration eye-tracking experiments. This allows us to create a common set of AoI labels between the members of a set of subjects, allowing direct visual comparison. The identified labels can be further examined using sequence analysis methods so that an analyst can identify similarities and differences in visual scanning behaviour amongst the subjects. Also, real-world eye tracking experiments are prone to noise due to users moving away from the field of view of the sensor or due to sensor errors. In our example application scenario, ~30% of the eye tracking data collected from an air traffic control simulation was noisy. Taking this into account, we have explored the use of sequence analysis approaches such as multinomial Hidden Markov Models [4] and a sequence mining application named ELOQUENCE [5]. Our method extends the approach developed in [2] to identify common AoIs labels across multiple users, which can be then utilized to analyze the similarities and differences in visual behaviour between all of the users. The main contributions of our approach are the following:

- Automatic identification of common AoI labels across multiple subjects with minimal user input.
- Analysis of user behaviour using multinomial Hidden Markov Model and sequence mining based approaches in the presence of noisy eye gaze data.

## 2. Related Work

Eye gaze points are labelled using two prominent approaches - Area of Interest (AoI) and Object of Interest (OoI) based methods. The distinction between these two approaches lies in what questions they aim to answer. AoI based approaches reveal how a user inspects and monitors different regions of the displayed scene, while OoI based approaches reveal which distinct objects in the scene attract user attention. AoI based approaches can have a greater level of detail than OoI, as the total number of OoI labels is tied to the number of distinct objects in the scene. We classify the related work section into three parts: (1) and (2): comparison of visual behaviour using OoI or AoI labelling, respectively, and (3): comparison methods without any labelling. For recent reviews of eye tracking data analysis and visualization readers can refer to [6, 7].

### 2.1. Comparison using OoI labelling.

Image processing algorithms were used in [8] to compute OoIs in images. Scale-invariant feature transforms [9] and spectral clustering [10] were used for automatic detection of OoIs in videos. Different visualization approaches such as transition graphs, 'scarf plots', and OoI timelines are then used to compare eye gaze behaviour between multiple users [11]. The focus of our work is to reveal how different regions of a scene (AoIs) attract visual attention providing a greater level of detail in visual attention labelling than OoIs.

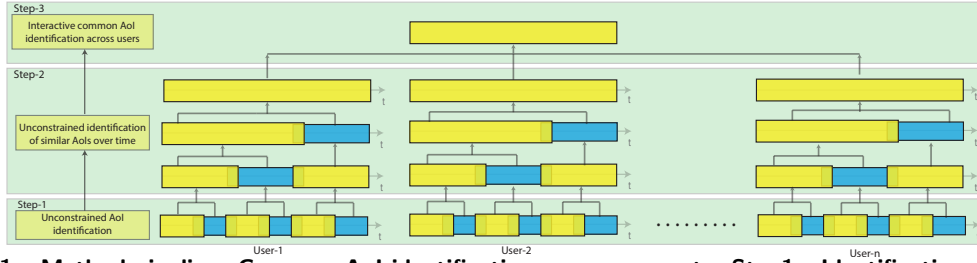
### 2.2. Comparison using AoI labelling.

Manual annotation of AoIs is very difficult for long duration eye tracking data due to the variation in behaviours which may occur. Automated approaches compute AoIs using pre-defined grids or data-driven methods. Predefined grid-based methods divide the scene into a user defined grid and gaze points falling within a grid cell are labelled with the cell id, but they can fail to capture the level of detail in the visual scanning behaviour. Labelling the AoIs based on spatio-temporal characteristics of the data provides a higher resolution in the identified AoIs and better captures the evolution of the users' attention over time. The ScanMatch [12] approach allows manual definition of AoI labels through simple binning of the scene into a discrete number of regular bins. It was extended in [13] using a percentile mapping approach. Data driven methods such as Mean-shift clustering were used in iComp [14] to cluster the fixation points into AoI labels. The fixation clusters are represented as ellipses and an intersection is performed across the ellipses to

label AoIs across users. Over et. al. [15] constructed Voronoi cells around fixation points based on their density, where the areas of high fixation density lead to small Voronoi cells and vice-versa. Unfortunately none of these methods handle the problem of data saturation, a common issue with very long duration eye tracking datasets [2], well. The Levenshtein distance [14] is used to compute a similarity score between two users but this approach cannot define the relationship between AoIs and how close they are in space and time. The Needleman-Wunsch algorithm [12] can be used to model relationships between AoIs using a similarity matrix but it has to be defined manually. Multidimensional scaling [16] can then be applied on these similarity matrices to identify groups of users exhibiting similar gaze behaviour. A more exhaustive state of the art in comparing AoIs between multiple users is available in [17]. Long duration eye tracking experiments tend to generate noise, due to sensor errors or users moving out of the field of view of the sensors, and the challenges in dealing with this uncertainty in the data are not discussed extensively in the previous methods. Probabilistic transition matrix based methods [18] compute the amount of transition between the AoI labels of the stimulus and can be used to compute the variability between a pair of users. However, the number of transition matrices across multiple time steps and users increases dramatically for a large dataset.

### 2.3. Comparison without labelling.

Geometric distance based comparison methods are used to map neighbouring fixations using Mannan distance [19] but the challenge is that several fixations from a single eye gaze sequence can be mapped to a single fixation in the other. This leads to overestimated similarity scores and hence a disproportionate impact of individual fixations. The MultiMatch algorithm [20] computes distances across multiple dimensions such as the shape of eye gaze samples, saccade length and direction, fixation position and duration and converts the data into a vector based representation. Similarity between users is statistically tested using the Kolmogorov-Smirnov test. Heatmaps [21] can be computed based on the density of fixation points and compared across users using similarity measures such as correlation coefficient, Kullback-Leibler distance, and earth mover distance. A state of the art discussion on available comparison methods and metrics is presented in Holmqvist et. al. [22]. Without the application of any data abstraction techniques such as data labelling or clustering, however, analyzing very large eye tracking



**Figure 1. Method pipeline: Common Aol identification across user sets. Step1 - Identification of fixation clusters. Step2 - Similar Aol identification for every user. Step3 - Common Aol identification across all users.**

datasets collected across multiple users hugely increases the computational complexity of the analysis process.

The main contributions of our proposed method are, (1) Computation of common AoI labels across multiple users using scalable clustering and cluster merging that is not constrained by the total number of parameters that need to be ‘tweaked’ by an analyst (Sections 3.1, 3.2, 3.3), (2) Comparison of user behaviour using two prominent sequence analysis approaches in the presence of noise in the eye gaze data (see section 3.4).

### 3. Our method

As discussed in Section 2, clustering methods for identifying AoIs in long duration eye tracking data suffer from data saturation. The work of [2] highlights this problem when applying Mean-shift clustering for long duration eye gaze data. To overcome this problem, they propose a clustering and hierarchical cluster merging to compute AoIs. The AoI labels are not, however, synchronized across all subjects and hence the comparison of visual scanning behaviour between subjects can be only performed manually. Also the method requires a number of parameters to be defined by an analyst and it becomes difficult if we want to compare behaviour between a large number of subjects. In order to avoid data saturation and to remove the difficulty around modifying multiple parameters, we propose the following list of improvements for labelling AoIs.

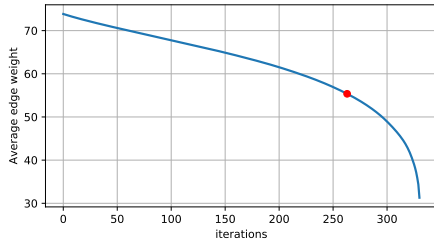
#### 3.1. Step1 - Clustering raw eye gaze data

For each subject, the raw eye gaze data with its timestamp information is divided into overlapping time windows. The earlier work of [2] used 2D Minimum spanning trees (MST) to cluster eye gaze points. Classification of eye gaze data into fixation clusters and saccades are performed based on the spatio-temporal characteristics of consecutive eye gaze points. Hence, clustering the data without temporal information can lead to mis-classification. This problem can, to some extent, be avoided by setting small

overlapping time windows but that approach introduces additional parameter tuning. Instead, we propose a 3D MST based clustering approach where the nodes of the tree represent 2D eye gaze data with its corresponding timestamp. The edge weight of the MST is computed using the Euclidean distance metric. Fixation clusters from the 3D MST can then be computed by deleting its inconsistent edges. As described in the earlier work of [2], Zahn’s clustering algorithm requires tuning of input parameters such as neighbourhood depth and inconsistency factor for each eye tracking data set. In order to avoid the complexity of tuning input parameters for each eye tracking data set, we extend the Maximum Standard Deviation Reduction (MSDR) algorithm of [23] to delete inconsistent edges in the 3D MST. Let  $\sigma(T)$  represent the standard deviation of all the edge weights of the MST. At each iteration,  $i$ , an edge is removed such that it maximizes the reduction of the weighted standard deviation  $\sigma(T^i)$  of the MST sub-graphs. If the edges of the MST sub-graph at the  $i^{th}$  iteration is  $T^i : \{T_1^i, T_2^i, \dots, T_m^i\}$ , the standard deviation of edge weights of the sub-graph be  $\sigma(T_j^i)$ , the  $\sigma(T^i)$  is computed using equation 1.

$$\sigma(T^i) = \frac{\sum_{j=1}^{|T^i|} |T_j^i| \cdot \sigma(T_j^i)}{\sum_{j=1}^{|T^i|} |T_j^i|} \quad (1)$$

The local minimum of a higher order polynomial fit on the difference between successive  $\sigma_i^2$  values corresponds to the optimal iteration to terminate the process (see [23]). Each MST sub-graph represents an eye gaze cluster and as the maximum duration of a saccade is 80ms, MST sub-graphs with less than  $\sim 5$  gaze points can be discarded as saccades for an eye tracker of refresh rate 60 Hz. The convex hulls of the MST sub-graphs can be represented using polygons and in the subsequent steps, only the polygons are considered for further processing. Any fixation classification algorithm can also be used in this step, but our approach removes the complexity of working with multiple input parameters.



**Figure 2. Significant drop in average edge weights of the MST is highlighted by the knee point in red.**

$$d_j(A, B) = 1 - \frac{\text{area}(A \cap B)}{\text{area}(A \cup B)} \quad (2)$$

### 3.2. Step2 - Hierarchical cluster merging

Spatially neighbouring AoIs that are separated in time may end up having different AoI cluster labels. In order to compute maximum spatial similarity among the AoIs, minimum spanning trees can be effective in solving this combinatorial problem. For every pair of adjacent time windows, we compute an adjacency matrix of the cluster polygons. Each entry of the adjacency matrix corresponds to the spatial similarity between a pair of cluster polygons and is computed using a symmetric Jaccard distance metric (see Equation 2). An MST is computed from this matrix, where the weight of an edge corresponds to the amount of dissimilarity between two cluster polygons. The longest edges in the MST correspond to cluster polygons of high spatial dissimilarity. The edges are sorted and removed in an iterative manner starting from the longest edge. The average edge weights representing the average value of the Jaccard distance metric can be plotted against the iteration step as shown in Figure 2. A knee point can be computed using [24] which indicates the iteration index at which there is a significant drop in the dissimilarity. This entire computation requires no user intervention and can be parallelized across every adjacent pair of time windows. Cluster polygons with high spatial similarity are merged together and this process of merging adjacent time windows can be repeated in an hierarchical manner, as shown in Figure 1.

### 3.3. Step3 - Common AoI labelling

The topmost time window of the hierarchy diagram (see Figure 1) contains cluster polygons representing areas of interest from all the subjects taking part in the same scenario of an eye tracking experiment. In order to compute common AoI labels across all the subjects, a similarity matrix is computed from all the cluster polygons as explained in Step 2. Cluster polygons

which exhibit strong similarity are merged based on the computation of the knee point. At this juncture, an analyst can also modify the knee point to control the degree of similarity while merging the clusters (see Figure 2). Moving the knee point towards the left leads to a higher degree of cluster merging leading to very few areas of interest and vice-versa.

### 3.4. Step4 - Comparison of multiple subjects

The common AoI labels from the topmost time window of the hierarchy (see Figure 1) can be propagated back to the lowest level containing the raw eye gaze points where each point gets labelled either as noise, saccade or an AoI label id. The raw eye gaze data is transformed into a symbol sequence for each subject and sequence analysis methods can be used to identify patterns of visual gaze behaviour across the subjects. We analyze the data with two prominent sequence analysis approaches, Hidden Markov Models and pattern-growth based methodology [25] for sequence mining, in order to explore subject behaviour across multiple subjects.

**Hidden Markov models (HMM).** Instead of inspecting the similarities in the recorded eye-tracking data directly, a probabilistic modelling approach which takes account of the uncertainties in the data due to noise in the eye-tracking sensors can be preferable. For categorical time series of AoI labels, an HMM [26] is a natural option to find a probabilistic description of the AoI patterns, both for smoothing noisy high-frequency data and for predicting missing observations (e.g. due to sensor malfunction). As each move between AoIs is typically separated by a saccade, a discrete-time HMM which only considers simple transitions between time points is insufficient as this would result in a model where each transition between two AoIs is through a ‘‘saccade state’’. While a continuous time HMM could be applied only for AoIs of interest, this approach would also rely heavily on potentially noisy local transitions. Hence, we consider the following multinomial HMM with aggregated time intervals. Our starting point is the aggregated counts of AoI labels during time intervals of length  $\Delta$ . By denoting the time interval  $[t, t + \Delta]$  simply as  $t$ , we define

- *Observed label set sequence*  $\mathbf{Y} = (Y_1, \dots, Y_T)$  with  $Y_t = (n_t^1, \dots, n_t^L)$ , where  $n_t^l$  is the number of observed labels  $l$  at time  $t$ .
- *Hidden state sequence*  $\mathbf{x} = (x_1, x_2, \dots, x_T)$  with hidden states  $s \in \{1, \dots, S\}$ .
- *Initial state probability vector*  $\pi = \{\pi_s\}$  of length  $S$ , where  $\pi_s$  is the probability of starting from the hidden state  $s$ .
- *Transition probability matrix*  $A = \{a_{sr}\}$  of size

$S \times S$ , where  $a_{sr}$  is the probability of moving from the hidden state  $s$  at time  $t - 1$  to the hidden state  $r$  at time  $t$ .

- *Emission probabilities*  $p(Y_t)_s, s = 1, \dots, S$ , where each  $p(Y_t)_s$  is a multinomial distribution defining the state specific emission probabilities of observations  $Y_t$ .

For a more detailed description of HMM see, for example, [4]. Given  $K$ , the number of hidden states and the observations  $\mathbf{Y}$ , we can estimate the initial probability vector, transition probabilities and multinomial distributions, either via maximum likelihood or by a Bayesian approach.

**Sequence mining.** The path of AoIs followed by the subject's eyes during an eye-tracking experiment can be seen as a sequence of events, where each event corresponds to a visit to an AoI. This makes it possible to use sequence mining techniques to investigate the subjects visual scanning strategies during an experiment. In contrast to the methods in sections 2 and Hidden Markov models, sequential patterns of eye movement across multiple subjects, which may be shifted by a time span, can be easily detected. Input for a sequence mining algorithm consists of a data set,  $D$ , of event sequences and a minimum support threshold, denoted by  $\sigma$ . In the concrete case of the work presented here, an event corresponds to an AoI and an event sequence captures the sequence of AoIs derived from eye movement data. A sequence  $\alpha \equiv e_1 \rightarrow \dots \rightarrow e_n$  is a *subsequence* of another sequence  $\beta \equiv e'_1 \rightarrow \dots \rightarrow e'_m$  ( $0 \leq n \leq m$ ), if  $e_1 = e'_{i_1}, \dots, e_n = e'_{i_n}$ , for some integers  $1 \leq i_1 < \dots < i_n \leq m$ . The output of a sequence mining algorithm is a set of AoI sequences  $\Theta$ , called frequent patterns. Each frequent pattern  $p \in \Theta$  occurs as a sub-sequence of  $k \geq 1$  input sequences and  $\frac{k}{|D|} \geq \sigma$ . In the context of the work described in this paper, sequence mining can be applied

- to investigate eye movement strategies common to multiple subjects who performed a common experiment, or
- to find patterns on the sequence of AoIs obtained from a single subject.

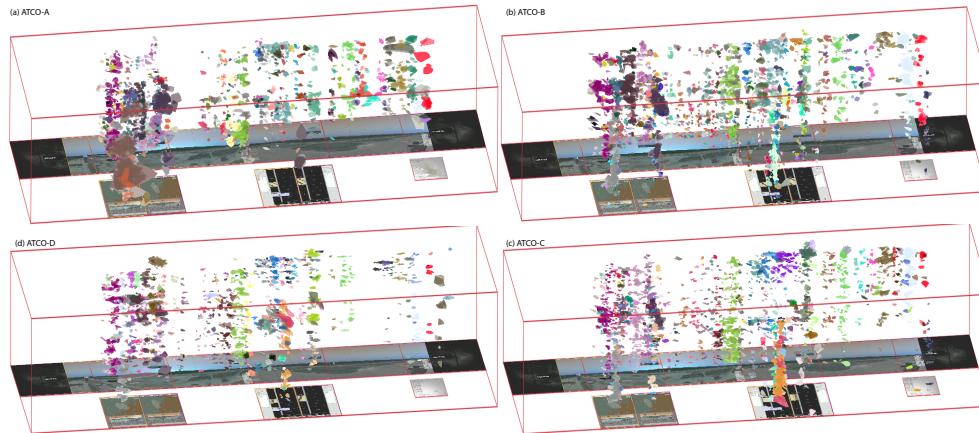
In the former case, it is important that a common alphabet is used to label the AoIs discovered for the multiple subjects. Thus, the computational flow starts by using the method described in section 3.3 for computing common AoI labels across different subjects, then sequences of AoIs corresponding to the gaze movements of each subject can be obtained by labelling the raw eye gaze points of each participant. A gaze point in the raw data that falls within an AoI  $X$  (i.e. a cluster polygon) is

labelled with the  $X$ 's label. Finally, sequence mining is used to find common eye movement strategies. For the latter case corresponding to a single subject, a sliding window over the input sequence of AoIs generated by the subject is used and a set of sub-sequences can be obtained from this window at different points in time. This set can then be mined to discover sequential patterns corresponding to frequent occurring eye-movement strategies employed by the subject while solving tasks such as monitoring aircraft landings and take off. The sequence mining algorithm used is based on a pattern-growth approach [25] and is part of the interactive visualization system ELOQUENCE [5].

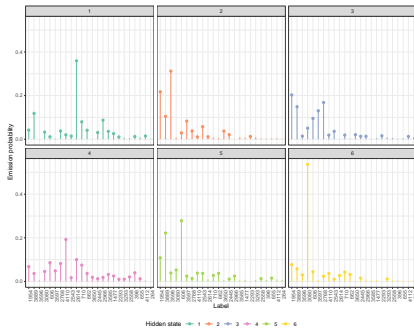
#### 4. Application Example: Simulation of Remote Multi-Tower Air Traffic Control Scenario

The source of the data for demonstration and testing of our methods are derived from a new way of working in the field of Air Traffic Control. The 'Remote Tower' approach has been in development for more than ten years and offers the opportunity to staff numerous smaller airports with a single controller. The traditional tower is replaced with cameras, and the Air traffic control officer (ATCO) works in a virtual tower where large display environments provide the equivalent of a tower view and can display more than one airport view at a time. This approach reduces the number of ATCOs required to provide national cover, reducing costs, but this new way of working presents significant changes in procedure and is currently a topic of thorough study to determine best practices and ensure safety of operation. The experimental setup presented in [3] is used for data collection and it consists of six large screens presenting the tower view for two airports. The screens on the desk present, respectively, radar information for both airports, flight 'strip' management for both airports, and a voice communication system VCS to control communication with the aircraft.

**Data collection.** Eye tracking data sets were collected from four operational air traffic controller officer (ATCOs) performing the same scenario. SmartEye [27] camera sensors were used to track the eye movements of the controllers in a non-intrusive manner. On average around 30% of the data samples exhibited uncertainty due to sampling errors and also due to controllers moving their attention away from any of the nine screens. During the experiment the controllers can move in a comfortable manner but are required to remain seated. The eye-tracking data is recorded in the local coordinate system (pixel values) of each of the screens and each experimental scenario



**Figure 3. Common labelling of AoIs across four controllers: Common AoIs are marked with the same colour in each sub-figure. The space-time cube (STC) can be rotated, zoomed and filtered along vertical axis to avoid visual clutter. The two tower views are displayed at the back of the cube while the radar, flight strip and the voice control displays are arranged across the front part of the cube.**



**Figure 4. Emission probabilities of the states identified using HMM across all users.**

lasted for  $\sim 90$  minutes for each controller. For the purposes of our analysis, the eye tracking data was remapped into a single space-time volume, as shown in Figure 3, as if the displays were parts of a single, larger flat display environment.

#### 4.1. Common AoI labelling across all users

For each user, the gaze data was divided into 1 minute time windows with 50% overlap. These values were chosen since any task performed by an ATCO such as assessing the flight, radar, runway etc., should require less than 30 seconds. As our method does not depend on user-defined parameters, Steps 1 & 2 require no intervention from the analyst and hence can easily be scaled to any number of subjects. Common AoI labels are computed across all the users in Step 3 and at this stage, an analyst can control the degree of similarity across AoIs by interactively adjusting the knee point. Moving the knee point towards the left in Figure 2 leads to a high degree of cluster merging, eventually producing less AoIs and vice-versa. The results for four ATCOs carrying out the same scenario

are displayed in Figure 3. All computations were implemented in Python on an iMac with a 3.5Ghz Intel Core i7 processor and 16GB RAM. Step1 takes between 15 to 20 minutes for a single user, but this processing needs to be done only once for each data set and advanced parallel processing libraries can be utilized to improve the performance. In Steps 2 & 3, finding intersections between polygons were computed using kd-trees, which makes the computation logarithmic, and the iterative edge removal requires logarithmic time to sort the edge weights.

#### 4.2. Hidden Markov Model (HMM)

In case of multinomial HMM, we chose 4.5 minutes time intervals, a duration chosen based on the time taken for any significant event, such as a take-off or landing, in the simulation, creating in total  $n = 19$  time windows for each user. While shorter time intervals can provide more fine-grained details of subject behaviour, longer intervals can better capture groups of commonly occurring labels (the estimated hidden states contain multiple large emission probabilities) and reduce the effect of small time-scale variation and missingness in the AoI data. As the interest was in comparing the four participants, we modelled a joint model for all four participants, leading to common estimates of  $\pi$ ,  $A$ ,  $B$ , and thus common definitions of the hidden states. Due to computational costs involved in identifying the parameters, using the full set of extracted AoIs was not feasible. Therefore we focused on AoIs which occurred commonly for all four users. We first extracted a reduced set of labels with a condition that the total dwell time of each label was at least two seconds for one or more users. This resulted in 370 AoIs. From these,

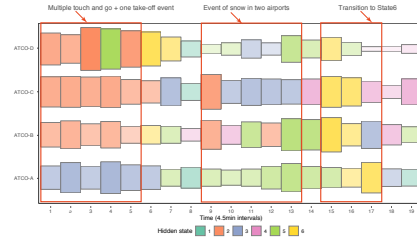
we used only those common labels across all the four users (intersection of users' reduced label sets), leading to 22 distinct labels.

The HMM was written in Stan [28], and the analysis and visualization was done in the R environment [29] using rstan [30] and ggplot2 [31] packages. A common problem in HMM framework is the definition of  $K$ , the number of hidden states. Too small or large number of hidden makes interpretation of the models as well as general model fit difficult. Small  $K$  leads to hidden states which combine large number of observed symbols, whereas increasing  $K$  produces typically simpler (sparser) emission patterns, but complex transition structures (see, for example, discussion in [32]). Note however, that with large  $K$ , possibly with sparsity constraints between states, HMM approach can be useful as a pre-processing method for creating reduced sets of aggregated labels, which can be further analyzed, for example, with a sequential pattern mining approach. As our goal was not prediction but exploring and comparing the behaviours of the participants, we chose  $K = 6$  based on visual inspections of the model parameters, hidden state paths, and the corresponding label patterns mapped against the simulation environment and observed sequences.

We used Dirichlet priors for  $\pi$ , rows of  $A$ , denoted by  $A^k$ , and rows of  $B$ , denoted by  $B^k$  as,

$$\begin{aligned} \pi &\sim \text{Dirichlet}(\alpha_1^\pi, \dots, \alpha_K^\pi), \quad \alpha_k^\pi = 2^{K-k}, \\ B^k &\sim \text{Dirichlet}(\delta, \dots, \delta), \quad \delta \sim \text{half-Normal}(0, 1) \\ A^k &\sim \text{Dirichlet}(\alpha_1^{A,k}, \dots, \alpha_K^{A,k}), \\ \alpha_j^{A,k} &= \begin{cases} \frac{1}{2}(1 + \frac{T}{K}), & \text{if } |j - k| = 1, \\ 1 + \frac{T}{K}, & \text{if } j = k, \\ 1, & \text{otherwise,} \end{cases} \end{aligned}$$

where  $T$  is the number of time points (19) and  $\delta$  is an additional concentration parameter. The prior on  $\pi$  concentrates initial distribution towards first states for identifiability (ordering of states is arbitrary), while the prior on  $A$  emphasises diagonal dominant transition matrices a priori, and small values of  $\delta$  emphasise sparse emission densities. Based on above definitions, we obtained the MAP (maximum a posteriori) estimates of our model parameters and most probable hidden state paths with Stan using L-BFGS-B optimization. Figure 4 describes the composition of each state in terms of emission probabilities. Figure 5 shows the most probable hidden paths for each user. The height and opacity of each rectangle is proportional to the label count using the common label set (number of labels within time-interval belonging to the set intersection of

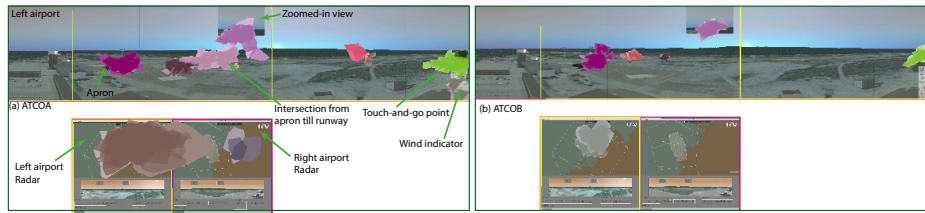


**Figure 5. State sequence for four controllers with information on contextual significance of each state given by colour intensity and size.**

all AoI labels) to the label count using the reduced label set (370, set union of all labels), with the maximum height corresponding to 62% and the minimum to 7%. We can see large similarities especially with ATCOs B and C, while ATCO D is spending relatively large proportion of time looking at something outside of the common label set. From the state sequence diagram of Figure 5, we can infer the common groups of observed AoI labels (labels which have large emission probabilities in same hidden state) and transitions between these label-groups (hidden states).

**Case 1: First 22 minutes of the simulation** From Figure 5, we see that in the first 22 minutes of the experiment, ATCOs B, C and D exhibit very similar behaviour, while ATCO A handles the scenario differently. Based on the available context data, only the left airport was active with multiple ‘touch and go events’ of an aircraft and a single take off event. ATCOs B, C and D are in State2, which exhibits concentrated attention at multiple parts of the airport as seen in Figure 6 (b). ATCO A remains in State3 where the main difference from other ATCOs is that there is a scanning behaviour at multiple parts of the airport with significant attention. Even the radar of the right airport receives major attention despite there being no traffic scheduled at that airport.

**Case 2: Snow event** Between time windows 9 to 13 (see Figure 5), we can observe different scanning behaviours between the ATCOs. From the context data, we find that there is a snowfall drastically reducing the visibility. During this time, a snowplough parked at a corner of the apron moves to clear the runway. During the snow event, there is also a take-off event in the left airport and a landing event at the right. ATCO A remains under State 5 for this entire time period, while ATCOs B and D remain in this state for most of the time. By observing the emission probabilities of State 5, the ATCOs exhibit concentrated attention at different AoIs as shown in Figure 7 (a). However, ATCO C remains in State 3 and, as explained above in Case 1, it corresponds to scanning behaviour at multiple parts of the airport, with significant time spent in each AoI to maintain the



**Figure 6. First 22 minutes of the experiment: (a) Major attention scanning pattern of ATCO A (b) Concentrated attention of ATCO B**

situational awareness. The event of snow in both the airports has modified the behaviour of ATCO C from a previously concentrated attention to a highly dispersed attention state of monitoring different AoIs.

**Case 3:** Between time windows 15 and 17 in Figure 5, all the ATCOs make a transition to State 6 that represents maximum visual attention at the right airport radar (see Figure 4). From the context data we find that there was a take-off event at the left airport and a landing-event at the right. Very high attention was paid by all ATCOs to the right airport radar to constantly update the position status of the incoming aircraft.

### 4.3. Common sequences between subjects

Exploratory sequence mining provides the facility for analyst-driven exploration of AoIs to identify patterns of interest in the data and also provides sequential information through the order of visits across AoIs. The four scanpath data sets from the ATCOs can be used as input for the sequence mining algorithm. The system used in this work is ELOQUENCE [5]. We first investigated the existence of eye-movement strategies common to all subjects. Thus, we set the support constraint  $\sigma$  to 100% and requested the system to compute sequence patterns of length 3. These patterns are shown in Figure 8(a). We were then interested in finding which AoIs were visited, by all four users at some point in time, after visiting AoI 3889. Thus, we have grown the tree branch, corresponding to the sequence pattern  $3080 \rightarrow 606 \rightarrow 3889$ . This leads to the conclusion that all four users used a “nested” back-and-forth scanning behaviour: after visiting AoI 3080 the users scan between AoIs 3889 and 1954 and, some time after, return to AoI 3080. These AoIs are shown in figure 8(b). Our next goal was to investigate common scanning strategies that occurred before and after visiting AoI 662. We used the forward and backward capability of our system to mine interesting patterns. After selecting an AoI, forward mining computes sequence patterns starting with that AoI, while backward mining computes sequence patterns ending with it. An example is given in Figure 8(c). In the graph shown in this Figure, each path (without

repeated edges) starting at node 662 corresponds to a sub-sequence found in the four input sequences of AoIs (i.e.  $\sigma = 100$ ) Finally, we refer to the possibility to set time constraints on both the dwell time of the AoIs and on the time elapsed between AoIs of the computed sequence patterns, in order to find more meaningful patterns. Hence, the simplicity of this search mechanism allows for a fine-grained exploration of related activities within the long-duration eye gaze data, finding frequent patterns which correspond with user behaviour. This same information also allows frequent patterns to be filtered out of the data, allowing infrequent patterns to be identified, meaning that the analyst can explore less frequent, even aberrant behaviours.

## 5. Discussion

By taking into account the characteristics of the eye tracking data, the proposed improvements removes the restrictions in tuning multiple parameters for AoI identification. The technique also allows an analyst to define the degree of similarity across the identified AoI clusters through modification of a knee point plot. Identification of an optimum knee point can then be handled by an analyst using an interactive visual analysis system. The advantage of this approach is that the identification of areas of interest is data driven and the total number of identified AoIs can be completely controlled by an analyst. With the increase in the number of identified Areas of interest, it enables more detailed analysis of the visual scanning patterns but it adds additional computational complexity for the sequence analysis methods.

By transforming the raw eye gaze data into symbolic representation, sequence analysis can be used to identify patterns in eye gaze behaviour. While different sequence analysis methods from bio-informatics and text mining communities are available for analyzing the symbol sequence, two prominent approaches using HMM and Sequence mining are utilized to identify patterns in the data set. HMM approach can be used to infer multiple groups of observed AoIs and transition of eye gaze behaviour between the groups of AoIs. But due to the computational complexity



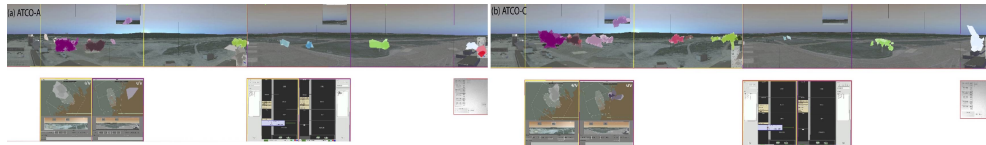


Figure 7. Comparison of ATCOs A and C during the snow event

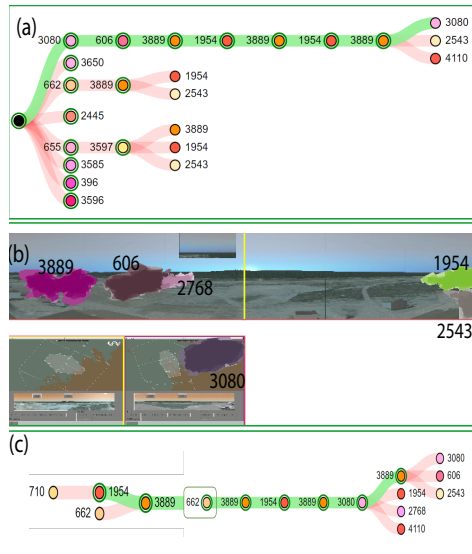


Figure 8. (a) Common eye-movements to all users. (b) AoIs identified in the shown sequences. Common eye-movements before and after visiting AoI 662.

of HMMs for analyzing large data sets from multiple users, interactive exploratory data analysis becomes difficult. Hence, HMM based approach can be useful as a pre-processing method for creating reduced sets of aggregated AoIs, which can then be further analyzed for example with sequential pattern mining approach. The sequence mining algorithm based on a pattern-growth approach [5] enables analyst-driven exploratory data analysis to identify patterns of interest. The order of visits to different AoIs and their frequency of occurrence across multiple users can be identified by an analyst using the interactive visual analysis system. Different sequence mining approaches from text mining and bio-informatics communities can be evaluated in the future to identify the effectiveness of different methods for tasks performed by an analyst. But in order to perform an evaluation of different sequence analysis techniques, ground truth information that represents the intention for a visual scanning behaviour can be collected from ATCOs, using for example, talk out loud methods with predefined code words indicating the intention of the actual activity.

## 6. Conclusions and Future Work

In this work we have developed a method which, with little or no input from an analyst, computes

common Areas of Interest (AoIs) labels across multiple users who are undertaking a similar or identical task. The common set of AoI labels permits the use of any number of methods to explore common and different behaviours between subjects. We have exemplified this through the application of sequence analysis methods such as Hidden Markov models and Sequence mining. The specific case described in this work relates to ATCOs performing experimental studies being carried out with respect to safe practice in remote tower scenarios, where the ATCO is simultaneously responsible for traffic at two or more different airports. Our colleagues in air traffic management, who have provided the data sets, have found these approaches very interesting, particularly for the long-duration experiments which they frequently conduct, and we have begun working to conduct an evaluation of the effectiveness of these methods in analysing data of this type in comparison with current techniques.

The HMM framework can also be extended in various ways. For example, the label occurrences and hidden state transitions can be explained through additional information, such as the timing of various events on the screen or the characteristics of the participants. In the case of a large number of participants, mixtures of multiple HMMs can be used to group participants into sub-populations of interest [33]. We are also currently studying the optimal combinations of the label set definitions and time interval lengths for various settings.

One aspect of eye-tracking data which must be dealt with when considering sequence mining is that the data is very noisy, with missing or erroneous samples due to occlusion, interference or other issues. Our initial version of sequence mining method has avoided this issue but not dealt with it directly. We plan to improve the quality of the sequences through better filtering and better use of the temporal information available.

## Acknowledgements

We thank Lothar Meyer and Supathida Boonsong of the Swedish Air Navigation service for providing the data sets and domain expertise. Funding for this research project was provided by the Swedish Transport Agency, Trafikverket, and the Swedish Science Council.

## References

- [1] W. Fuhl, T. Kuebler, T. Santini, and E. Kasneci, "Automatic generation of saliency-based areas of interest for the visualization and analysis of eye-tracking data," in *Proceedings of the Conference on Vision, Modeling, and Visualization*, pp. 47–54, 2018.
- [2] P. K. Muthumanickam, K. Vrotsou, A. Nordman, J. Johansson, and M. Cooper, "Identification of temporally varying areas of interest in long-duration eye-tracking data sets," *IEEE TVCG*, vol. 25, no. 1, pp. 87–97, 2019.
- [3] P. K. Muthumanickam, A. Nordman, L. Meyer, S. Boonsong, J. Lundberg, and M. Cooper, "Analysis of long duration eye-tracking experiments in a remote tower environment," in *13th USA/Europe air traffic management R&D seminar, Vienna, Austria*, 2019.
- [4] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.
- [5] K. Vrotsou and A. Nordman, "Exploratory visual sequence mining based on pattern-growth," *IEEE TVCG*, vol. PP, pp. 1–1, 06 2018.
- [6] G. Andrienko, N. Andrienko, M. Burch, and D. Weiskopf, "Visual analytics methodology for eye movement studies," *IEEE TVCG*, vol. 18, no. 12, pp. 2889–2898, 2012.
- [7] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl, "Visualization of eye tracking data: A taxonomy and survey," in *Computer Graphics Forum*, vol. 36, pp. 260–284, 2017.
- [8] L. Paletta, K. Santner, G. Fritz, A. Hofmann, G. Lodron, G. Thallinger, and H. Mayer, "A computer vision system for attention mapping in slam based 3d models," *arXiv preprint arXiv:1305.1163*, 2013.
- [9] T. C. Kübler, D. R. Bukenberger, J. Ungewiss, A. Wörner, C. Rothe, U. Schiefer, W. Rosenstiel, and E. Kasneci, "Towards automated comparison of eye-tracking recordings in dynamic scenes," in *EUVIP, 2014*, pp. 1–6, IEEE, 2014.
- [10] K. Kurzhals, M. Hlawatsch, C. Seeger, and D. Weiskopf, "Visual analytics for mobile eye tracking," *IEEE TVCG*, vol. 23, no. 1, pp. 301–310, 2017.
- [11] T. Blascheck, M. Schweizer, F. Beck, and T. Ertl, "Visual comparison of eye movement patterns," in *Computer Graphics Forum*, vol. 36, pp. 87–97, Wiley Online Library, 2017.
- [12] 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.
- [13] T. C. Kübler, E. Kasneci, and W. Rosenstiel, "Subsmatch: Scanpath similarity in dynamic scenes based on subsequence frequencies," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 319–322, ACM, 2014.
- [14] A. T. Duchowski, J. Driver, S. Jolaoso, W. Tan, B. N. Ramey, and A. Robbins, "Scanpath comparison revisited," in *Proceedings of the 2010 symposium on ETRA*, pp. 219–226, ACM, 2010.
- [15] E. A. Over, I. T. Hooge, and C. J. Erkelens, "A quantitative measure for the uniformity of fixation density: The voronoi method," *Behavior research methods*, vol. 38, no. 2, pp. 251–261, 2006.
- [16] 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.
- [17] T. C. Kübler, C. Rothe, U. Schiefer, W. Rosenstiel, and E. Kasneci, "Subsmatch 2.0: Scanpath comparison and classification based on subsequence frequencies," *Behavior research methods*, vol. 49, no. 3, pp. 1048–1064, 2017.
- [18] S. R. Ellis and J. D. Smith, "Patterns of statistical dependency in visual scanning," 1985.
- [19] 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*, 1996.
- [20] 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.
- [21] R. Caldara and S. Mielliet, "iMap: a novel method for statistical fixation mapping of eye movement data," *Behavior research methods*, pp. 864–878, 2011.
- [22] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.
- [23] Y. Zhou, O. Grygorash, and T. F. Hain, "Clustering with minimum spanning trees," *International Journal on AI Tools*, vol. 20, no. 01, pp. 139–177, 2011.
- [24] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a" kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, pp. 166–171, IEEE, 2011.
- [25] J. Han, J. Pei, and X. Yan, "Sequential pattern mining by pattern-growth: principle and extensions," *Studies in Fuzziness and Soft Computing*, pp. 183–220, 2005.
- [26] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.
- [27] Smart Eye AB, Gothenburg, Sweden, "Smart Eye Pro."
- [28] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of Statistical Software, Articles*, vol. 76, no. 1, pp. 1–32, 2017.
- [29] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.
- [30] Stan Development Team, "RStan: the R interface to Stan," 2018. R package version 2.18.2.
- [31] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.
- [32] S. Helske, J. Helske, and M. Eerola, *Combining Sequence Analysis and Hidden Markov Models in the Analysis of Complex Life Sequence Data*, pp. 185–200. Cham: Springer International Publishing, 2018.
- [33] S. Helske and J. Helske, "Mixture hidden markov models for sequence data: The seqHMM package in R," *Journal of Statistical Software, Articles*, no. 3, pp. 1–32, 2019.