

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/327052318>

# Identification of Temporally Varying Areas of Interest in Long Duration Eye Tracking Data Sets

Article in IEEE Transactions on Visualization and Computer Graphics · August 2018

DOI: 10.1109/TVCG.2018.2865042

---

CITATIONS

0

READS

73

5 authors, including:



Prithiviraj Muthumanickam  
Linköping University

5 PUBLICATIONS 4 CITATIONS

[SEE PROFILE](#)



Katerina Vrotsou  
Linköping University

29 PUBLICATIONS 209 CITATIONS

[SEE PROFILE](#)



Jimmy Johansson  
Linköping University / Norrköping Visualization Center C

60 PUBLICATIONS 698 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Master Thesis Project [View project](#)



Stations environments and suicides [View project](#)

# Identification of Temporally Varying Areas of Interest in Long-Duration Eye-Tracking Data Sets

Prithiviraj K. Muthumanickam, Katerina Vrotsou, Aida Nordman, Jimmy Johansson and Matthew Cooper

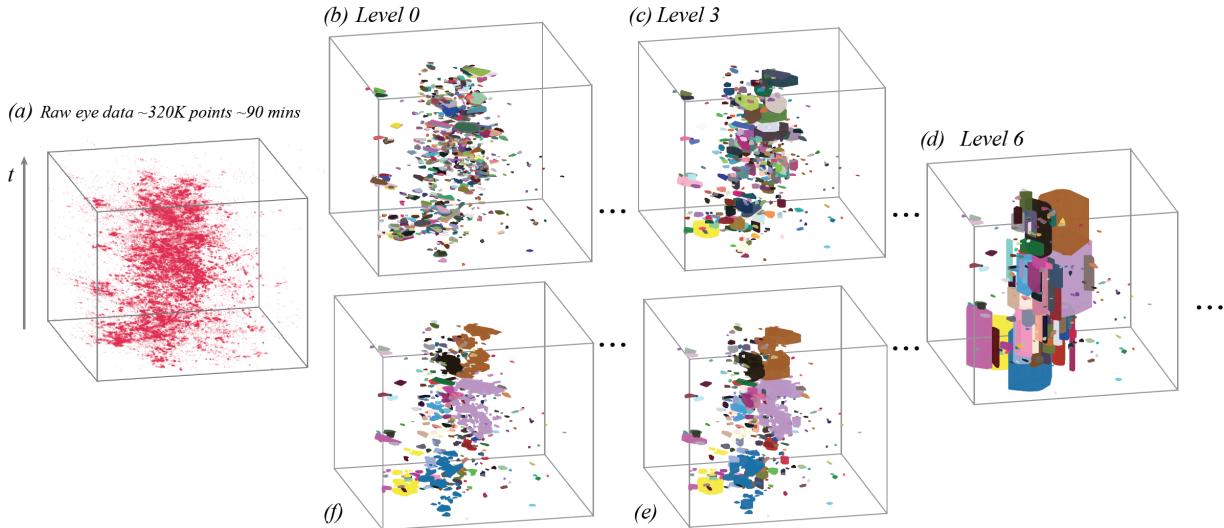


Fig. 1: Finding patterns of spatial visual attention in long-duration eye-tracking data (here  $\sim 90$  minutes) suffers from visual clutter as shown in (a). We sub-divide the eye-tracking data into overlapping time windows and perform clustering within each to identify areas of spatial visual attention. A cluster merging approach is then used to create a hierarchy of cluster levels. This enables the analyst to explore the changing patterns of attention which may exist over several time windows, disappearing and reappearing as the task progresses. (b) results of clustering the eye-tracking points into 1 minute time windows with 50% overlap at level 0. (c) and (d) merging the clusters into 8. and 32.5 minute time windows at levels 3 and 6, respectively. In (f) and (e) the clusters of levels 0 and 3 are redrawn but coloured according to the clusters in level 6. This allows the analyst to identify recurrent AOIs and observe the dynamic shape of each.

**Abstract**—Eye-tracking has become an invaluable tool for the analysis of working practices in many technological fields of activity. Typically studies focus on short tasks and use static expected areas of interest (AOI) in the display to explore subjects’ behaviour, making the analyst’s task quite straightforward. In long-duration studies, where the observations may last several hours over a complete work session, the AOIs may change over time in response to altering workload, emergencies or other variables making the analysis more difficult. This work puts forward a novel method to automatically identify spatial AOIs changing over time through a combination of clustering and cluster merging in the temporal domain. A visual analysis system based on the proposed methods is also presented. Finally, we illustrate our approach within the domain of air traffic control, a complex task sensitive to prevailing conditions over long durations, though it is applicable to other domains such as monitoring of complex systems.

**Index Terms**—Eye-tracking data, areas of interest, clustering, minimum spanning tree, temporal data, spatio-temporal data

## 1 INTRODUCTION

Eye-tracking experiments typically focus upon tasks which have a duration of no more than a few minutes or seconds. The data acquired can, therefore, be readily explored to identify patterns of subjects’ attention using simple static visual representations, such as heat maps, and temporal change can be observed using a simple animation. When dealing with longer duration experiments, however, extracting behaviour is made much more complex due to clutter in the raw data itself, and identifying patterns of change made near impossible due to the many potential time frames which must be compared. The main motivation

of this work is, therefore, to develop techniques that support the analyst in the exploration of tasks performed by subjects in very long duration (hours) eye-tracking experiments, for which standard methods have proven too difficult and take too long to be practical. These qualities are present in many long-duration monitoring tasks such as examining the attention patterns of car drivers, the monitoring of large and complex systems (e.g. industrial processes, power distribution), traffic flows, and even the behaviour of audiences attending theatre or sporting events. The authors have come to this problem due to the desire to conduct experiments in the field of air traffic control (ATC), where controllers behaviour must be observed for a period of 1-2 hours, and their patterns of behaviour will change not only due to changes in the traffic flow but also as a consequence of changes in prevailing conditions, such as weather or unexpected events (such as in-flight emergencies or communication failure).

All authors are with the Linköping University. <prithiviraj.muthumanickam, katerina.vrotsou, aida.vitoria, jimmy.johansson, matthew.cooper>@liu.se  
Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.  
Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

The problem presents a set of features which differ from most eye-tracking studies. First, the data is of long duration, presenting problems of cluttering and with the identification of changes in behaviour over time. Second, while the subject’s focus is driven by objects in the

display, the subject does not specifically watch the objects. Indeed, in the case of an air traffic controller (ATCO), they are trained to never focus on any single object for a significant length of time. Instead, they develop a sense of ‘situational awareness’ and maintain this by looking at regions of the display at suitable times to update their understanding of the situation. This is similarly true for the driver of a vehicle. Third, the processes being monitored are driven by more than a single simple display, but may also involve attention paid to features outside of the display such as, in the case of a driver or tower ATCO, looking out of the windows or looking at other support information such as paper documents, other information displays or system controls. Consequently, the behaviour of the subject may not be captured by analysing what lies in the display and must be identified directly from the eye-tracking data, and so the focus of the subject’s attention over time. The problem also presents an additional aspect in that the subject is involved in a live and long-duration task, where they are controlling the events. For example, an ATCO is directing the aircraft under their control and so, where there exists more than one way to resolve a possible issue during their work, the subsequent behaviour of the system under observation may diverge between two sessions, even when the initial conditions of the system are identical. What behaviour of the subjects leads to a different opinion about the best solution is something that may be captured by comparison between the subjects’ separate behaviours over time. These problems raise a number of design goals for the work we present here, each to meet a specific need in the analysis:

**G1** **An algorithmic process for the identification of temporally varying areas of interest** in long-duration eye-tracking experiments, where the aim is to understand *how* the visual attention of a subject evolves over time. The process could be fully automatic or controlled by the analyst as appropriate.

**G2** **A system for interactive visual analysis** providing a summary view of the identified AoIs over the entire duration of the data. The system should support the identification of important characteristics of the subject’s behaviour and the identification of changes in behaviour at potentially unanticipated points.

**G3** **Identification of similarities and differences** between individual subjects’ behaviour despite variation in their actions leading to differences in the progress of the task.

This work addresses the goals presented above by using a combination of clustering of the raw data, temporal merging of the clusters to produce a hierarchy of AoIs and their display in an interactive Space-Time Cube (STC) [23] representation. The STC representation allows the analyst to examine the temporal evolution of AoIs as they appear, disappear and recur over time in long-duration experiments. A visual comparison can also be performed to identify any substantial variation in the behaviour of a set of subjects undertaking the same experiment. The main contributions of our approach are the following:

- A time-dependent clustering and cluster merging method to identify spatial AoIs that appear, disappear and change over time.
- A visual representation of long-duration visual attention behaviour through the display of changes in the identified AoIs over time.
- Flexible navigation of the hierarchy of identified AoIs using information from higher levels to aid the analyst in tracking recurrent behaviour.
- A scalable approach that can handle very long-duration eye-tracking data through parallelization of the data analysis.

## 2 RELATED WORK

Visual attention of subjects is commonly studied using eye gaze data collected through eye-tracking experiments. Visual attention can be broadly classified into two types: bottom-up and top-down attention

[15, 34, 39]. Bottom-up attention refers to involuntary attention, where a salient stimulus can attract attention even though the subject had no intentions to attend to it. Top-down attention refers to voluntary allocation of attention, where subjects focus their attention to: (1) a small well-defined region of space in the scene (top down spatial attention) referred to as an Area of Interest (AoI), or (2) a moving or otherwise distinct object in the scene (top down feature attention) referred to as an Object of Interest (OoI). The similarity in top-down and bottom-up approaches is that, although the reason for visual attention may be different, the attended stimuli receive preferential processing and hence there will be an accumulation of eye gaze points around them in the collected data. Identifying and labelling these accumulated eye gaze points across a scene and analyzing them can help identify eye movement patterns, cognitive load and subjects’ behaviours.

An important distinction to make between the two types of top-down attention (AoIs vs OoIs) lies in the questions they aim to answer. The identification and study of spatial AoIs aims to reveal *how* a subject inspects and monitors a scene which may contain moving or distinct objects. This contrasts with OoIs identification which aims to reveal *what* distinct or moving objects in the scene are monitored by the subject. Readers can refer to [26, 29, 30, 38, 40] for methods that identify OoIs in eye tracking data. Though OoIs can capture the attention shift between distinct objects in the scene, their associated spatial context is lost along with their temporal evolution (in case of moving objects). In our work, we are interested in identifying spatial AoIs in long duration eye tracking datasets. In order to identify the context that attracted the spatial attention, we perform an in-context visual analysis of AoIs by overlaying them over the video capture of eye gaze data. In the following, we discuss the present state of the art on various methods for identification and visual analysis of AoIs and their disadvantages in dealing with long-duration eye-tracking data sets are also highlighted.

### 2.1 Identification of AoIs

Areas of interest change over time with respect to their size and position during long-duration eye-tracking experiments. AoIs of a first time subject looking at a certain stimulus can be different from an experienced subject looking at the same stimulus. Hence, there is a time evolving nature to AoIs and labelling them can be useful in understanding *how* a particular subject is paying attention over a stimulus. In general, manual identification and labelling of AoIs is a difficult task [12] and several approaches are studied for automating this process. We classify identification and labelling of AoIs in eye-tracking data into two main categories: 1) methods using manually predefined grids, and 2) automated methods.

#### 2.1.1 Manually predefined AoIs

A common approach to the identification and labelling of AoIs is to divide the scene into predefined grids, or a set of manually defined areas, and then assign appropriate labels to eye data points falling within them. The resolution of the grid in this case determines the accuracy and the number of identified AoIs [11, 20]. The disadvantage of predefined static AoIs is that they can fail to capture the temporal evolution of the subjects’ attention over time, particularly when long-duration experiments are involved. Consequently, such methods, do not adhere to our design goals **G1** and **G2**. In these cases, it becomes a necessity to adopt other methods for identifying and labelling the AoIs.

#### 2.1.2 Automated methods

Several approaches for automatic identification of AoIs have been suggested that are mainly based on clustering of the raw eye-tracking data. As discussed earlier, AoIs are often varying over time and hence clustering which takes time into account needs to be applied to understand gaze movement strategies [6]. The research available on this is extensive but not well suited to long-duration data. In fact, the identification of time evolving AoIs in stimuli that contain either active or passive content is ranked as important future work in a recent survey [8].

Traditional approaches dealing with analytics of eye-tracking data start with eye movement classification algorithms to differentiate between fixations and saccades. They involve threshold and bayesian

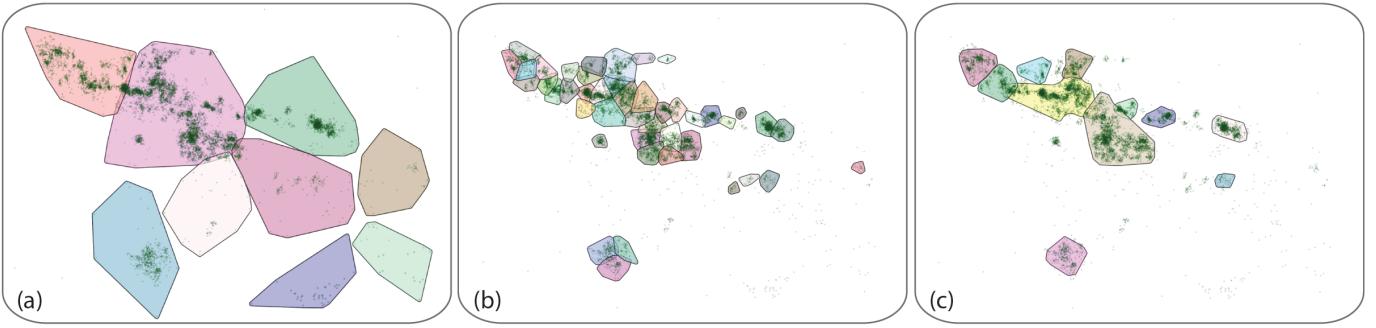


Fig. 2: An example of identifying AOIs using a mean shift clustering and our own method on a 4.5 minute time interval of a data set composed of 2D eye gaze points collected at 60Hz. Mean shift applied with (a) high (0.15) and (b) low quantile (0.01) values results in either too little and too much separation in the identified AOIs, respectively. (c) Temporally varying AOIs identified using our method display more balanced results.

learning based methods for eye movement classification. For example, Tobii [1] employs the Velocity-Threshold Identification (I-VT) algorithm. Grouping the consecutive fixations into cluster hulls can generate AOIs over time. Since the classification algorithms are not completely accurate, if there is a misclassification of a saccade as a fixation point, it can lead to erroneous clusters. Also, clusters identified in this way within a short time duration can end up with different cluster labels. Methods such as Mean shift [17, 22, 24, 27, 42] and Gaussian Mixture Models [35, 44] have been used to perform clustering on the whole eye-tracking data set to identify AOIs. These methods do not satisfy any of our design goals because they do not consider the changing behaviour of an observer over time. Therefore, they are not suitable for direct application on long-duration data sets as when the amount of eye gaze data increases, the local AOIs get distorted or saturated resulting in very few clusters (AOIs) that were accumulated over time.

Methods that consider eye-tracking scan paths as trajectories have also been used for analysing eye-tracking data and identifying AOIs [16]. The work of Andrienko et al. [3] provides a thorough overview of visual analytics approaches from Geographic Information Science (GIS) with their merits and drawbacks for analyzing eye movement data. We tried a trajectory clustering algorithm [4] that considers both 2D spatial information of the gaze points and the time stamp value to find time evolving AOIs. We found, however, that using such an approach did not produce satisfying results primarily due to the nature of the eye gaze trajectories and also due to the long-duration of eye-tracking recordings. Unlike the trajectories of moving objects, eye gaze points are not spatially continuous over time, since eye gaze data commonly involve frequent attention switching over the stimulus. This is particularly true in the case of air traffic controllers' visual scanning patterns. Smooth pursuits are an exception as they generate a continuous trajectory of data when a subject is continuously tracking a moving object. However, our goal (as stated in **G1**) is to identify how attention to spatial AOIs evolves over time and not to identify and track actual moving objects in the scene (OOIs). Furthermore, the percentage of smooth pursuits in the entire eye-tracking data set depends on the nature of eye-tracking task. In the specific case of air traffic control, smooth pursuits are considered a dangerous behaviour because they can lead to a loss of local situational awareness. Overall, the methods described above are not suitable for direct application to long-duration experiments because they are mainly focusing on extracting AOIs from the entire eye gaze data set projected onto a 2D space, which becomes problematic as the data increase and local AOI information gets lost.

While the curse of multi-dimensional data is its data sparsity in higher dimensions, we are faced with high accumulation of data in 2D over time that distorts the information. We exemplify this by applying Mean Shift clustering [14] on a 4.5 minute time interval of a data set composed of 2D eye gaze points collected at 60Hz, as shown in Figures 2(a) and (b). In the example, *high* (0.15) and *low* quantile (0.01) values were used leading to time varying AOIs getting labelled at either a very high or very low granular level, respectively. We tested with various quantile values in between but were unable to find any that produced results comparable with our approach, where we build AOIs from the smallest time windows to larger ones by merging the

AOI clusters based on their strength of intersection.

## 2.2 Visual analysis of eye-tracking data

Eye tracking data is collected in the form of gaze position and time stamp. Visualization of time oriented data is discussed extensively in [2]. Blascheck et al. [7] categorizes visualization techniques for eye tracking data into 2D and 3D, animated and static, spatial, temporal and spatio-temporal, in-context and not in-context, etc.

2D Heat maps, or attention maps, is one of the simplest visualization technique, where areas with differing eye fixation density can be mapped to a colour scale. Such maps can be constructed in real-time [18] and are used to aggregate the fixation data over the stimulus and mark out areas with high accumulation of eye gaze points. Methods presented in [21] and [48] quantify the similarity of eye fixation points and the degree of coverage of fixations over a visual stimulus. While they can provide information about the distribution of eye fixation density over the stimulus, they fail to display any temporal characteristics of the identified AOIs. Another disadvantage is that they can easily saturate the entire stimulus with their colour scale, when long-duration eye-tracking experiments are involved. As we are dealing with long duration eye tracking datasets, small multiples based visual representation can suffer from lack of screen space while animation based visualization techniques are not suitable due to difficulties in preserving the viewers mental map [2]. This makes such methods not applicable to our problem as per our design goal **G2**.

Space-Time Cube representations can capture variation over space and time in the same view which is why they have been used in several studies [5, 33]. In case of eye-tracking data, they can display the time evolving nature of AOIs, their temporal duration, underlying stimuli (dynamic/static), and aggregated information from multiple subjects [16, 31]. Due to these characteristics of the STC, we use it along with 2D spatial projection of time slices from STC. Kurzhals et al. [28] propose an approach that combines the use of STC with synchronized scarfplots and timeline views in order to identify and represent AOIs/OOIs over time. The main difference between this work and our approach is that they focus on the identification of objects over time in the scene, while we are interested in identifying interesting spatial AOIs over time (as per **G1**). Due to this, the use of scarfplots and timelines is inappropriate for us because these representations lack the spatial reference which is central to our work. Moreover, previous approaches focus upon tasks which have duration of no more than few minutes, while our method deals with eye gaze recordings in the order of hours.

In general, many pure visual representations such as AOI trees, AOI rivers, graphs, scarfplots [9, 10, 13, 32, 45] fail in combining both the spatial and temporal characteristics of the AOIs and OOIs. Furthermore, they often suffer from visual clutter, when dealing with long-duration eye-tracking data. In order to overcome these shortcomings, a semi-automatic analyst driven data mining approach for interactive exploration of AOI sequences over time was proposed in [36]. They proposed to adapt algorithms used for exploring sequences in a large event-based data [46]. Along with data mining and analyst driven interactive exploration of AOIs, only those AOIs of interest to the analyst can be visualized over the stimulus, thereby reducing visual clutter

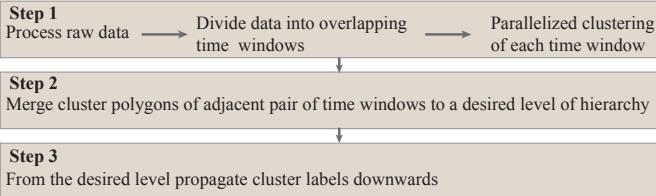


Fig. 3: Process pipeline for identification of temporally varying AoIs

in long-duration eye-tracking data sets. As a first step of the above proposed visual analytics model, our aim in this work is to identify temporally varying spatial AoIs in long-duration eye-tracking data sets (**G1**), which can then be further explored using data mining approaches, as described in [36, 46, 47].

### 3 OUR METHOD

The automated approaches for the extraction of AoIs discussed in Section 2.1.2 are not suitable for direct application to long-duration eye-tracking experiments due to the high accumulation of data in 2D over time which distorts the local AoI information. Our proposed method for identifying temporally varying AoIs is composed of the following 3 steps which are described in detail in subsections 3.2-3.5.

1. Divide the long-duration eye-tracking data into overlapping time windows and run a clustering algorithm (Section 3.1) independently on each time window using parallel threads (Section 3.2).
2. Merge clusters from neighbouring time windows in a hierarchical manner, based on the strength of overlap between neighbouring clusters. Repeat the merging to the desired level of hierarchy (Section 3.3).
3. From the desired level of hierarchy, propagate cluster labels downwards (Section 3.4).

Implementation and performance details of the complete system are described in Subsection 3.6. The process pipeline of our method is shown in Figure 3.

#### 3.1 Zahn's clustering approach

Having explored a variety of algorithms, both the clustering and merging steps in our approach have been implemented using Zahn's Minimum Spanning Tree (MST) based clustering algorithm [49]. Before we present details of each step of our method, we briefly describe this graph-based clustering algorithm.

Consider a data set of points  $S$  in a metric space with a distance function  $\rho(x, y)$ . The data set  $S$  implicitly defines a weighted undirected graph  $G$  obtained by associating the cost  $\rho(x, y)$  with each edge  $(x, y)$ , for any two points  $x, y \in S$ . In [49], it is shown how the MST for graph  $G$  can be used to capture the cluster structure in  $S$ . The intuition behind the algorithm is to detect clusters in the data set by deleting those edges from the MST that have a significantly higher cost than the other edges. These edges are called inconsistent edges. The connected sub-graphs of  $G$  obtained after the removal of inconsistent edges form the clusters.

To determine the inconsistency of the edges, Zahn proposes several methods. The following was used in our work. Assume  $N_d(x)$ , with  $d > 0$ , denotes the set of graph edges in a path of length  $0 < i \leq d$  from node  $x$ . In other words,  $N_d(x)$  is the set of edges in the neighbourhood of node  $x$  within a depth of at most  $d$ . Consider also the average,  $\mu_{xy}$ , and the standard deviation,  $\sigma_{xy}$ , for the edges in  $N_d(x) \cup N_d(y)$ , for a given  $d > 0$ , and then determine the ratio  $r_{xy} = \frac{|\rho(x,y) - \mu_{xy}|}{\sigma_{xy}}$ . An edge  $(x, y)$  is inconsistent if this ratio is above a given threshold  $f > 0$ , called the inconsistency factor.

#### 3.2 Step1: clustering raw data points

In the first step of our proposed method, we pre-process the data for any null values in the eye gaze co-ordinates and replace it with a predefined spatial point in order to retain the information on when the subject

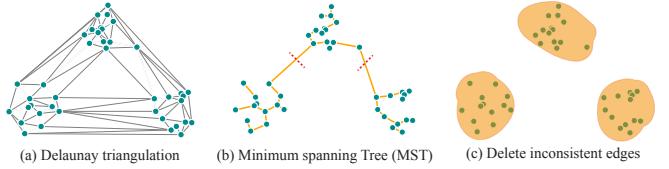


Fig. 4: Clustering of eye data points using Zahn's method [49].

moved away from the field of view of the eye tracker. The raw data points are then divided into overlapping time windows and clustering is applied to the points in each time window. The size of the time windows and amount of overlap is chosen by the analyst depending on the nature of the dataset. Following this, we use the Minimum Spanning Tree based clustering algorithm proposed in [49] to cluster the points within each time window. One important advantage of this method is that it can handle very large data sets, since clustering is performed independently in each time window and hence can be parallelized. Previously, MST based techniques [41] were used to identify fixation and saccades in eye-tracking data due to their degree of control and flexibility. However, their computational cost was high, which we overcome in our method through the use of parallelization for the cluster computation.

The nodes of our weighted undirected graph are eye gaze points and the initial graph is obtained by computing the Delaunay triangulation (DT) [19] of the eye-tracking points, as shown in Figure 4(a). The cost associated with an edge  $(x, y)$  corresponds to the Euclidean-distance between the nodes  $x$  and  $y$ . Note that an MST of the weighted undirected graph, implicitly defined by a set of points in a metric space with a distance function, is a sub-graph of DT. An MST is then computed (figure 4(b)) and inconsistent edges detected and removed using the method described in Section 3.1. The resulting clusters correspond to the connected sub-graphs. The output of the first step of our method is the set of cluster polygons (Figure 4(c)) corresponding to these connected sub-graphs.

The results of this step correspond to the first and most detailed level of granularity of the identified AoIs and will, in the following, be referred to as level 0 of our approach. The clustering results can be displayed in a space-time cube [23], as shown in Figure 1(b). The STC's horizontal plane corresponds to spatial coordinates, in this case spatial positions of eye gaze points, and the vertical axis corresponds to time going upward. We choose the Zahn's method [49] for our clustering step because of the degree of control, flexibility [41] and for uniformity, since the same process is used in the second step of our method. Other clustering algorithms could, if appropriate, be used in this first step to cluster the raw data.

#### 3.3 Step2: cluster merging AoIs

A drawback with performing clustering over time windows, as outlined earlier, is that spatial visual attention over a stimulus evolves over time and hence a visual attention cluster (AoI) can spread across adjacent time windows. This would eventually lead to different cluster labels, as the clustering in a single time window is independent of the rest. We overcome this problem by combining the clusters from a set of neighbouring time windows based on the strength of local overlap of the cluster polygons. This merging process can be carried out in a hierarchical manner which creates a spatio-temporal hierarchy, as shown in Figure 5. This hierarchy makes it possible to effectively assign meaningful labels to the identified AoIs over time, even if they are significantly separated in time. The merging process is performed in two steps and is repeated for each adjacent pair of time windows.

1. Compute an adjacency matrix of cluster polygons based on a cost function between cluster polygons.
2. Compute an MST from the adjacency matrix and apply the Zahn clustering algorithm to find cluster polygons with maximum intersection. The nodes of the MST represent the cluster polygons, while an edge between two cluster polygons  $x$  and  $y$  is associated with the cost between  $x$  and  $y$ .

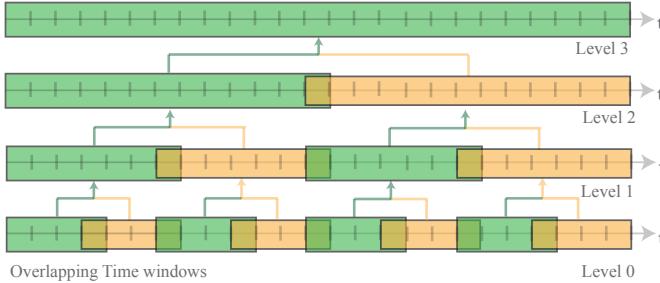


Fig. 5: The eye-tracking data is divided into overlapping time windows for clustering/temporally varying AoI identification. Cluster merging is performed in a hierarchical manner to identify a proper sub-division of time that does not saturate the dynamically changing AoI information. Computations at each time window in a level are independent of each other and hence can be done in parallel.

There are two main advantages to the proposed merging step. First it enables us to find local overlapping AoIs without clustering of the entire data. Second, the process is parallelizable and so re-computing the merging process at selected nodes in the hierarchy can be done in real time and under the control of the analyst.

### 3.3.1 Adjacency matrix of intersection costs

We compute an adjacency matrix to establish the neighbourhood relations between cluster polygons from two consecutive time windows. The adjacency matrix,  $C$ , is a symmetric  $N \times N$  matrix where  $N$  is the total number of cluster polygons from two consecutive time windows. The entries of the adjacency matrix range from 0.0 to 1.0. The value  $C(i, j)$  represents how strongly the polygons  $i$  and  $j$  are connected with each other, as shown in equation 1. If there is a high overlap between two cluster polygons (see Figure 6(a),(b)) then there is a minimal cost and vice-versa. The entry  $C(i, j)$  is 0.0, if there is no intersection between polygons  $i$  and  $j$ . Otherwise, the value of the cost function is in the range  $[\epsilon, 1.0]$ , where the value of epsilon  $\epsilon$  corresponds to two polygons with full intersection.

$$C(X, Y) = \begin{cases} 0 & , X \cap Y = \emptyset \\ \min\left(\frac{\text{area}(X - Y)}{\text{area}(X)}, \frac{\text{area}(Y - X)}{\text{area}(Y)}\right) & , X \cap Y \neq \emptyset \wedge X - Y \neq 0 \wedge Y - X \neq 0 \\ \epsilon & , X - Y = 0 \vee Y - X = 0 \end{cases} \quad (1)$$

An additional parameter for temporal decay can be included in the cost function to control the merging of two cluster polygons based on their temporal proximity. For our example data, however, the chosen cost function (Equation 1) works well without any temporal decay parameter. We perform  $N^2$  computations for finding intersections between polygons. While this is a costly operation its computation time in our example data is negligible because the number of polygons in each time window is small and decreases further at higher levels. Spatial data structures such as kd-trees could be used to accelerate intersection tests between polygons if their numbers are large.

### 3.3.2 MST-based clustering

There may be multiple intersections between cluster polygons (see Figure 8(a)) within a local spatial area due to the volatility of eye movements. The aim is to find maximum connectivity/intersection among the polygons. Minimum spanning trees can be effectively used to solve such combinatorial problems (see, for example, Figure 7). They are already used to remove intersections among nodes in a graph layout algorithm [37]. We construct an MST from the adjacency matrix computed from the previous step. Each node in the MST represents a cluster polygon and an edge represents the intersection between the polygons. The weight of an edge or edge length, corresponds to the cost of intersection between two polygons. The higher the cost the longer the edge between the two nodes. Thus the set of interconnected polygons with maximum intersection are closely connected in the MST, while connections with high cost are separated by longer edge lengths.

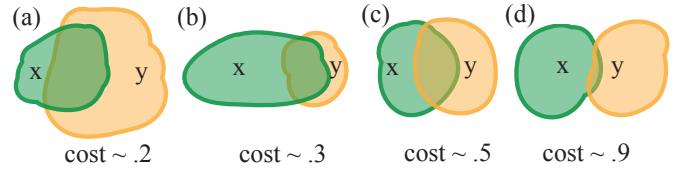


Fig. 6: Cost function to compute the strength of connection between two cluster polygons. (a) 20% of the area of  $x$  remains to be merged with  $y$  and so the cost is .2. (b) & (c) 30% and 50% of the area of  $y$  remains to be merged with  $x$ . (d) 90% of the area of  $x$  remains to be merged with  $y$  and so the cost is 0.9.

We use Zahn's method [49], described in Section 3.1, to find clusters of intersecting polygons. Finding inconsistent edges is performed in the same manner as in Step 1 of our approach (Section 3.2). Thus, the deletion of inconsistent edges leads to MST sub-graphs that correspond to AoI clusters.

### 3.4 Step 3: label propagation

The spatio-temporal hierarchy of levels shown in Figure 5 can be explored by the analyst to view the temporally evolving AoIs. The analyst can select a particular level of granularity from which assigned AoI labels (and colours) can be propagated downwards to shorter time windows, as shown in Figures 1(e) & (f). This helps to retain the temporal detail of the identified AoIs but assign a more coarse labelling to them which better reveals the subject's attention concentrations and shifts. This feature will be crucial when extracting AoI sequences from the data. These sequences can then be used to automatically identify visual scanning patterns through data mining algorithms.

### 3.5 Method parameters

The parameters used in our method are described below.

**Size of time window,  $t_{ws}$ :** The size of the time window is application specific and depends on the nature of the data. In general, it is advisable to use small duration time windows to avoid accumulation of data over time and let the merging process at higher levels control the granularity of the cluster polygons/AoIs. Since there is frequent attention switching in our example data set we set this parameter to a low value of one minute to avoid saturation of local AoIs.

**Percentage of overlap,  $t_{ov}$ :** Overlapping time windows are necessary to prevent AoIs in neighbouring time windows from being assigned different cluster labels. Increasing the overlap percentage increases the number of time windows at level 0 and vice-versa. This value is application specific and we tried with several values for our example data set and found an overlap of 50% to be suitable.

**Neighbourhood depth,  $mst_d$ :** Inconsistent edge detection requires the computation of the ratio  $r_{xy}$ , for each edge  $(x,y)$ , by exploring the edge's neighbourhood to a depth of  $mst_d$ . A high value for the neighbourhood depth would lead to few inconsistent edges and, consequently, a small number of clusters. The value should be heuristically determined based on the nature of the data set.

**Inconsistency factor,  $f$ :** The inconsistency factor value is also application specific. In our approach, the ratio  $r_{xy}$  is computed for every edge,  $(x,y)$ , of the MST and the values are displayed in a line plot to help the analyst to select an appropriate value to control the granularity of the clusters, as shown in sub-figures 8(b) and (c).

**Minimum overlap threshold for cost function,  $c_{min}$ :** A threshold for minimum overlap between two AoIs is included in the merging step. This allows the analyst to control the strength of the connection between two AoIs in order for them to be considered as one.

**Minimum number of points in a cluster polygon,  $cluster\_points$ :** The clustering of raw eye-tracking data discards the saccade points based on a minimum point threshold. Since the maximum duration of a saccade is 80ms [25, 43] and the refresh rate of the eye tracker used to collect our example data is 60 Hz, the number of points constituting a saccade can be  $\sim 5$ . In our algorithm, we safely initialize the minimum point threshold to be more than twice the saccade threshold 12. This value is set based on the refresh rate of the eye tracker.



Fig. 7: Merging different polygons based on maximum connectivity among them is solved using minimum spanning trees.

**Minimum area of cluster polygon,  $cluster_{area}$ :** We can set a threshold to decide a minimum area for the cluster polygon to be considered an AoI. This value is application specific and can be controlled by the analyst in order to be able to remove low surface area clusters, by treating them as noise.

In our method some parameters can be set to an initial value based on the nature of the application. Size of time window, percentage of overlap, neighbourhood depth are set once for our example data set taken from an air traffic control experiment. Parameters such as minimum overlap threshold, minimum number of points and minimum area of cluster polygon are used to filter the AoIs based on the requirements of the analyst. Inconsistency factor is used to control the granularity of the identified clusters, where increasing or decreasing the value can lead to an increased or decreased number of clusters (see Figure 8). The analyst is required to choose this value to control the localization of visual attention. For our example data set from the domain of ATC, we provide an initial value to control the granularity, but the analyst can change it in real time to meet their needs.

### 3.6 Implementation details and performance

Visual display of the eye-tracking data and cluster polygons, together with video playback of the stimulus were implemented in Qt and OpenGL. All computations were implemented in Python and multi-threaded where appropriate. We tested our approach with nine different data sets collected from ATCOs monitoring a radar screen. The data was recorded at 60 Hz which generated data sets of 270 k–320 k samples. We tested the performance on an iMac with a 3.5GHz Intel Core i7 processor, 16GB RAM and an NVIDIA GeForce GTX 775M graphics card with 2 GB memory. The computation time for our example data sets was between 33 to 52 seconds for calculation of the Delaunay triangulation, MST construction, computing cluster polygons and identification of inconsistent edges.

If  $N$  is the number of gaze points in a data set, and  $k$  is the initial number of time windows then the Delaunay triangulation, used in step 1 of our method, can be computed in  $O(N \log N)$  time and produces  $O(N)$  edges. Computing the ratio  $r_{xy}$  for each edge requires  $O(N^2)$  time. Consequently Zahn’s clustering algorithm also requires  $O(N^2)$  time. However, the computation time can be reduced by pre-computing and storing the neighbourhood MST of the raw eye data points. Clustering the raw data points is the most time consuming step in our method, since Delaunay triangulation followed by Zahn’s clustering algorithm are applied to each time window. However, this process can be parallelized over time windows. Calculating the distance between the nodes for cluster merging (step 2), requires  $O(N \log N)$  time. The height of the hierarchy built in step 3 is logarithmic on  $k$  and so each level in the hierarchy has  $O(k)$  time windows, each of which involves only a few identified AoIs and so is completed in milliseconds for our data sets.

## 4 APPLICATION INTERFACE

Our application interface (see Figure 9) provides an in-context visual display of the eye gaze data overlaid on the captured video of the stimulus and allows the analyst to browse the hierarchy obtained through the cluster merging, as shown in Figure 5. Analysts can also modify the results of clustering to their desired level of granularity in real-time.

**In-context visualization.** The 2D context visualization window (Figure 9(b)) displays captured video of the stimulus of the experiment in the background and can be overlaid with raw eye gaze points and/or the identified temporally varying AoIs. This view gives a contextual spatial overview of the AoIs on the screen and aids the analyst in understanding the stimulus’s events that attracted the visual attention.

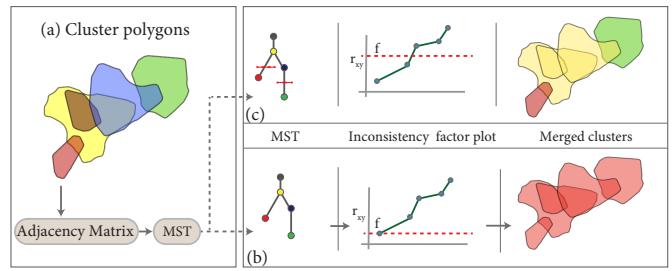


Fig. 8: Cluster polygons from adjacent pairs of time windows are merged based on maximum connectivity/intersections among them. (a) Adjacency matrix of cluster polygons are computed based on a cost function and the MST is computed from the adjacency matrix. (b) & (c) Inconsistency factor plot for the current MST can be tuned interactively to adjust the granularity of the clusters.

To capture the temporal aspects of the data, the main context window is supplemented with a 3D Space-Time Cube (Figure 9(a)) that displays eye points and/or temporally varying AoIs spread over time. In this view, the horizontal plane corresponds to the 2D stimulus, while time is represented on the vertical axis, increasing from bottom to top.

**Data exploration.** An interactive control panel composed of traditional widgets (Figure 9(c)) is available for the analyst to browse through the data. Analysts can browse the time windows at any level of the hierarchy and also navigate across the multiple levels of hierarchy as shown in Figure 5. By moving up and down the multiple levels of the hierarchy an analyst can identify the maximum level of data saturation that creates only a few big clusters which cover the entire area of the stimulus. At this point, the analyst can traverse down the hierarchy to their desired level of cluster saturation. They can browse through the time windows at that level of the hierarchy to find temporally evolving AoIs. Cluster polygons from the currently selected time window are displayed in the context visualization window, as shown in Figure 9(b). Furthermore, the control panel allows the analyst to select AoI colours from a particular level and have them propagated down to lower levels of the hierarchy. Four sliders are available to browse the data at desired time intervals, where the first two sliders are used to set a time interval, the third to position the time interval and the fourth to browse within that time interval. Based on the interaction of the fourth slider, the background of the Context window is updated by image frames from the video capture of the stimulus.

**Tuning clustering parameters.** Among the parameters described in Section 3.5, the inconsistency factor,  $f$ , controls the granularity of the AoI clusters, as shown in Figure 8. An interactive 2D plot (Figure 9(d)) is used to display the ratio  $r_{xy}$  to help the analyst to interactively set the inconsistency factor used for finding inconsistent edges, as described in Section 3.2. This plot allows analysts to set a maximum threshold by moving a horizontal line up and down on the plot, as shown in Figure 9(d). We can browse through the time windows at any desired level of the hierarchy (see Figure 5) and, once the threshold is set, the cluster merging process is re-run for the corresponding time window and all the higher levels of the time window hierarchy are updated accordingly. Parameters such as minimum overlap threshold of two cluster polygons, ( $c_{min}$ ), minimum number of points in a cluster polygon, ( $cluster_{points}$ ), and the minimum area of a cluster polygon, ( $cluster_{area}$ ), all set through the Control Panel (Figure 9(c)), provide high flexibility to analysts for setting the properties of valid spatial AoIs, depending on the application domain. The Control panel also allows the analyst to set the size of the time windows ( $t_{ws}$ ), the percentage of overlap between consecutive time windows ( $t_{ov}$ ), and the neighbourhood depth ( $mst_d$ ) used to determine the inconsistent edges.

All the views in the interface are linked and any changes or selections made by the analyst in the control panel will be reflected in all windows. Upon choosing a time window, data corresponding to it will be displayed in all views. Eye data points and/or identified temporally varying AoIs, belonging to the selected time window, will be shown in the main visualization panel and the STC. The visibility of these can be interactively toggled by the analyst through the control panel.

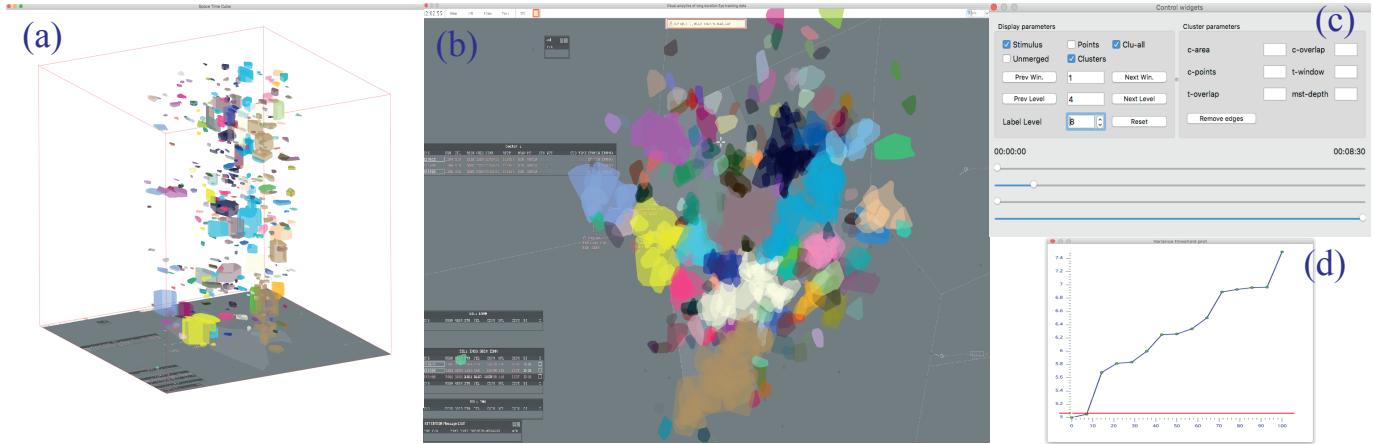


Fig. 9: The main views of our application interface: (a) Space-Time Cube displaying temporal aspects of the data and AoIs, (b) main 2D visualization window showing the raw data and/or AoIs, (c) interactive control panel, (d) interactive 2D plot displaying the ratio  $r_{xy}$ , for each edge  $(x,y)$ . An Image frame from the video capture of the stimulus is displayed as a background in (b) and as the base of the space-time cube in (a), and is updated as the analyst moves through the temporal data using the sliders available in (c).

## 5 APPLICATION EXAMPLE: SIMULATION OF REAL AIR TRAFFIC CONTROL SCENARIO

In this Section we demonstrate the results of applying our method to data collected from two operational air traffic controllers (ATCOs) conducting a training session on a simulation of flights into a medium-sized airport. ATCOs are required to do this type of training during their entire professional career. The performed training session was concerned with control of an air traffic sector through a radar screen with which the ATCO could interact via a mouse. After finishing each scenario they, together with an instructor, go through the scenario to discuss how the ATCO performed. Both ATCOs in this example conducted the same training scenario, which allowed us to compare their behaviour. This particular training session lasted for around 85 minutes, which is common for these types of training sessions. Training scenarios can be constructed with purely normal traffic, usually for trainee controllers, and with unusual events for more experienced controllers. Unusual events can include prearranged proximity events, failure of pilots to obey instructions, radar and radio failures, unidentified aircraft entering the airspace, unusual weather conditions. Tower simulations, which involve large screen displays to provide a panorama view of the airfield, can include issues with ground vehicles, pilot errors while taxiing or sudden poor visibility due to fog or very heavy cloud. Understanding how the ATCOs behave when dealing with these scenarios, their focus of attention, and to which features they may not be paying sufficient attention, is of great potential benefit to the trainer.

**Data collection.** While each ATCO was monitoring the radar screen (Figure 11), their eye movements were collected using a Tobii [1] eye tracker at a rate of 60Hz in an ideal laboratory environment. The radar screen content itself was also captured to video to make it possible to overlay the eye-tracking data on the dynamically changing radar display. ATCOs continuously switch their attention across different moving aircraft and update various flight settings such as heading or altitude either by voice commands to human ‘pilots’ in another room, who are controlling the simulation or by a simple computer interface within the radar screen as in our case.

**Identification of AoIs.** The collected data contains 3D gaze points as temporal positions on the radar screen  $(x, y, t)$ . We applied our method to the entire training session of  $\sim 85$  minutes. As per our **Step1** (see 3.2), the initial clustering of raw data points is performed using 1 minute time windows with a 50% overlap, as described in Figure 5. This window size was chosen since any action by an ATCO (assessing a flight, comparing it with the trajectory of another flight, or making an adjustment to a flight), should require less than 30 seconds. The clustering of the 320k samples is thus broken down into around 170 time windows, each of around 3600 samples. This generates our granularity level 0 in the hierarchy of merged clusters. The level 0 clustering removes very small clusters of eye points meaning that saccade information

is removed leading to a quite small number of well defined clusters for each window at granularity level 0. Since the cluster computation in each time window is performed in parallel, the process is more efficient (see Section 3.6) in terms of processing time. The merging process, **Step2** (see 3.3) was then applied for every pair of adjacent time windows leading to granularity level 1 containing 85 time windows of 1.5 minutes duration each. This merging process is repeated to level 2 with 43 time windows of 2.5 minutes duration each and so on to the highest level, in this case level 8, containing a single time window with the results of cluster merging for the whole data set. Initially, default values as described in Section 3.5 were used for clustering parameters. We browsed through level 0 of the hierarchy using our interactive control panel (Figure 9(c)) to view the results. We tested various values for the inconsistency factor using the inconsistency factor plot (Figure 9(d)) as described in Figure 8 before settling upon a value of  $f = 3$  for level 0 (eye data points). Among all the collected data sets that were used to test our method, we found this value to be sufficient to identify clusters with a high density of gaze points. For all higher levels we found  $f = 10$  to be suitable, but in some cases where we felt the need for refinement of the clustering results, we updated the value of  $f$  using our plot, as shown in Figure 8(b) & (c). The inconsistency factor relates to the type of data being investigated, hence it is regarded as an analyst-controlled parameter.

### 5.1 Air traffic Controller A

We begin our investigation by browsing the lower levels of the hierarchy of identified AoIs. Figure 10(a) shows one time window at level 5, corresponding to the first 16 minutes of the recorded data, together with the identified AoIs for ATCO-A. Based on our in-context visualization, we identify that these AoIs correspond to the attention of ATCO-A on aircraft moving along two routes, from the top and bottom left of the display towards the centre of the screen. We quickly identify interesting but expected results such as that the ATCO frequently focusses on the entry points, where new flights appear on their display. This is to identify the new flight and to understand its characteristics. Typically less attention is paid to each flight as they continue towards the airport unless a flight’s heading needs adjustment to avoid a proximity event. Significantly more attention is paid when the flights approach the glide path to the runway. We could confirm these findings from the first 16 minutes of the recorded video that showed aircraft approaching the airport via two different routes on the radar screen. Image frames from the captured video are displayed in the background of the 2D context window and on the base of the STC and controlled using a slider in the control panel. During these 16 minutes, the attention of ATCO-A switched between the two routes approaching the airport at the centre of screen but less attention was focussed on the airport since no aircraft had yet reached the glide path entry point. For the next 16 minutes, from

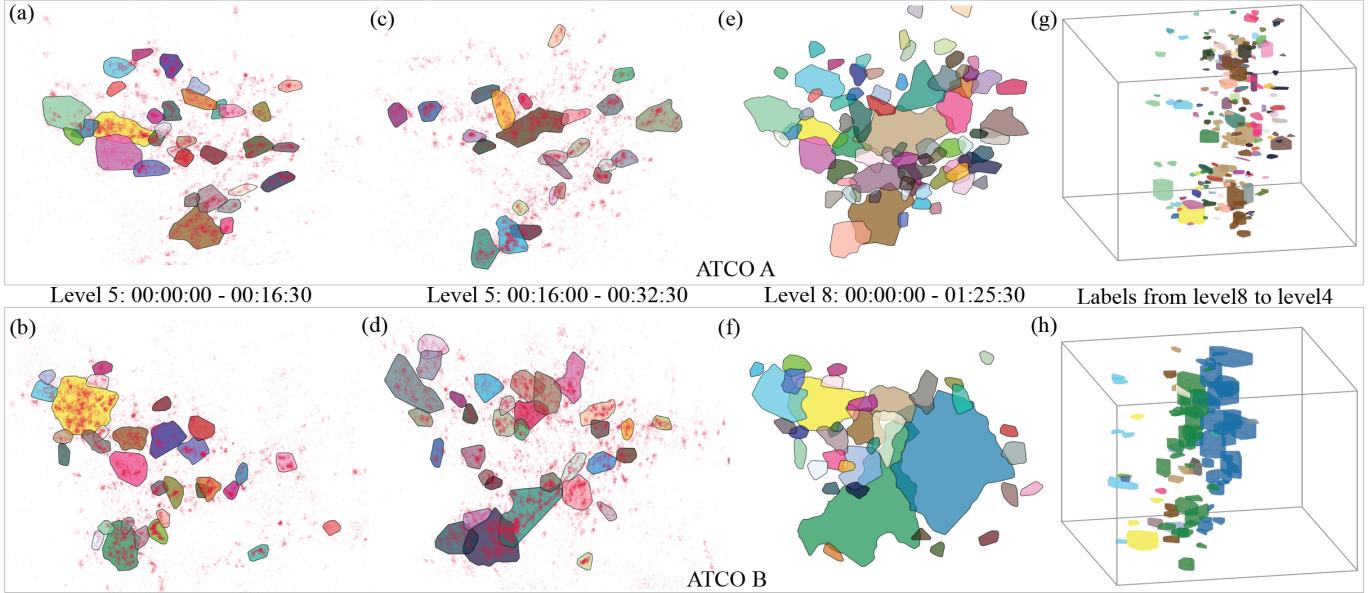


Fig. 10: Identification of AoIs for two ATCOs working on the same training scenario, lasting  $\sim 85$  minutes. (a) and (b) show the AoIs detected for the first 16:30 of the data for each ATCO at level 5 of our merged hierarchy, while (c) and (d) show the same result for the second 16 minutes of the data. Clearly the behaviour of ATCO-A has changed and their focus is more sparse across the display while that of ATCO-B remains similarly dense. (e) and (f) show the AoIs identified in the entire data set at the highest level in the hierarchy and the difference in behaviour is clear in sub-figure (f) where few, large AoIs remain. The STC representations (g) and (h), with colours propagated down from level 8 to level 4, make this difference in behaviour clear, as well as showing that there is a change in behaviour by both ATCOs roughly a third of the way through the scenario, in this case due to an abrupt influx of aircraft increasing the traffic density within the ATC sector.

16:30-32:30, similar AoI patterns were repeated when new incoming aircraft entered the display but more attention is now focussed on the airport at the centre of the display since flights have now begun arriving at the glide path. This can be seen in Figure 10(c). The interactive control panel facilitates the identification of these interesting (though expected) recurring patterns by browsing through our hierarchy. From these results we observe that our AoI identification approach identifies temporally varying AoIs which would have been otherwise lost when using traditional methods as discussed earlier (see Figure 2).

At higher levels we observe a large ‘blob’ at the centre of the radar display, corresponding to the airport, together with a more dense coverage of the display by the identified AoIs. Interestingly, ATCO-A exhibits a scanning strategy where their AoIs are typically fixed at certain geographical locations and there exists a clear recurring pattern of following the aircraft at different time intervals. At the topmost level of the identified AoIs we found an interesting pattern where there were clear regions on the screen that received repeated attention over the course of the entire training session (85 minutes). In this data set, the highest level (level 8) portrayed a good overview of the AoIs for the entire duration of the training session (see Figure 10(e)) and hence according to Step3 (see 3.4), the cluster labels were propagated down using the control panel from level 8 to level 4 as shown in Figure 10(g).

## 5.2 Air Traffic Controller B

The same training scenario was performed by ATCO-B. We begin our investigation by moving through the lower levels and, at the same time, analysed the recordings in a similar manner as for ATCO-A. For the first 16 minutes of the data recording, the attention of ATCO-B was switching between the two routes of the aircraft along their paths toward the airport at the centre of the screen. ATCO-B spent the next 16 minutes monitoring further incoming aircraft, as shown in Figure 10(b) & (d). In the second of those example images, (d), the density of AoIs along the flight paths is clearly much higher, however, than for ATCO-A. This indicates ATCO-B is looking at the incoming flights far more frequently in this second time period. At level 6 in the hierarchy, where the size of the time window is 32 minutes, the temporally varying AoIs became close to saturated. A condition which only worsens at higher levels, as shown at the highest level in the hierarchy in Figure 10(f).

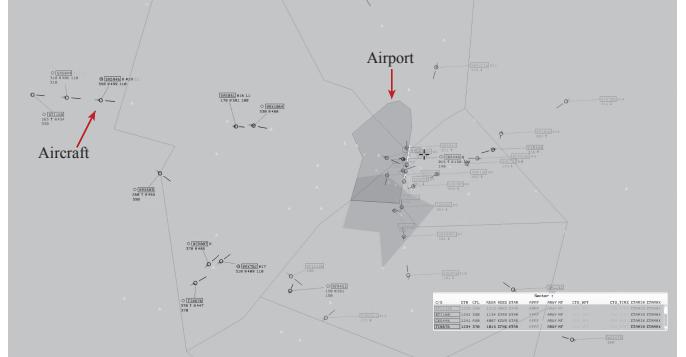


Fig. 11: Radar display showing flights with lead lines representing their heading. Additional information about each flight (type, speed, altitude and rate of climb/descent) is displayed in a box adjacent to the aircraft.

At these higher levels the information is lost and, hence, when using the 2D display, we can be restricted to working within shorter time windows at the lower levels of the hierarchy. The STC, however, has no such restriction and we can clearly see that our earlier hypothesis, that ATCO-B was revisiting each aircraft more frequently than ATCO-A, is correct. When examining the Figures 10(g,h) it is plain that the very large AoIs seen at the higher level are due to frequent visits to the different flights. For example in the green AoI of ATCO-B (Figure 10(h)), the STC display allows us to see that the merged AoI is created by many parallel AoIs moving with the flights as they are visited repeatedly. The STC display for ATCO-A shows a more sparse observation of the flights under their control, which leads to the better defined AoIs at higher levels. These differences show that ATCO-B has a very different strategy for maintaining ‘situational awareness’ of the flights under their control. They rely on rapid reacquisition of information while ATCO-A, relies more on memory. This may indicate different characteristics between the two ATCOs such as level of confidence, experience or, simply, quality of memory for details.

It is interesting to note that the STCs for the two ATCOs show greater similarity in the first third of the scenario run-time, with comparable levels of sparsity. After that point, however, the STC for ATCO-A

remains similar in sparsity at the periphery but with greater focus on the airport. ATCO-**B**, however, begins to show greater focus on all parts of the airspace. One possible cause is that the number of aircraft has increased at this point and ATCO-**A** has continued to cope well with this increased traffic density, while ATCO-**B** is under increased pressure and may be coping less well. In summary, the hierarchical structure of merged AoIs in our system has allowed us to easily explore these long duration eye tracking data sets and identify the differing characteristics of ATCO-**A** and ATCO-**B**. It has allowed us to examine these differences between their working strategies in context and even to identify what may be a weakness in one of the two ATCOs.

## 6 DISCUSSION

The problem of interactive exploration of long-duration eye-tracking data presents some unique challenges which have been addressed through the developments described in this paper. In the introduction a number of goals for the work were laid out. This section describes how these goals have been met and to what degree.

### **Goal G1 - Identification of temporally varying areas of interest.**

We process the raw eye-tracking data to determine the spatial areas which are of interest to the subject at different times throughout the duration of the lengthy session and to be able to classify and simplify those areas so that the analyst can understand how the patterns of the subject's attention change over time. Through the application of the hierarchical merging of low-level clusters derived directly from the data the analyst is able to see which areas are active at which points in the session. By selecting colour information from a higher level in the hierarchy the analyst is able to see those areas which, although sometimes active infrequently and at well-separated periods in time, are strongly spatially related and so relate to the same activity. Examples seen in our case of the air traffic control scenario are the entry points for the aircraft, the areas where aircraft are marshalled towards the runway and the points where flight trajectories typically intersect en route to the airport or to the exit points from the sector.

### **Goal G2 - Visual analysis.**

The aim of Goal G2 was to provide a clear summary view of the important characteristics of the subject's behaviour while carrying out the task in question. This has clearly been met in the STC representation of the identified Areas of Interest which allows an immediate grasp of the ATCO's activity throughout the session. This is clearly demonstrated in the case study where the two STCs showing the behaviour of the two ATCOs carrying out the training scenario are shown to be so distinctly different and where the analyst is then able to explore the temporal shifts between different patterns of behaviour and compare between the two. Using the interactive control panel, the analyst can browse through different levels of the hierarchy and through the time windows within a hierarchy level to identify differences in a subject's behaviour between different periods of time during the session. We can also explore and compare the subject's behaviour without prior knowledge of events which are expected to cause behavioural changes, making the exploration entirely data driven and allowing changes due to unanticipated events to be more easily identified. The system implemented permits exactly that as the display in the STC allows the analyst to identify shifts in subject behaviour and then examine the periods before and after those shifts in a 2D display that is overlaid on the background stimulus. The addition of simple interaction methods to clamp the display to shorter periods of time around the identified shifts in behaviour might be employed in more dense displays where clutter becomes more of an issue.

### **Goal G3 - Comparison of multiple subjects.**

The aim of this goal was to enable comparison between different subjects to identify similarities and differences in their behaviour when undertaking identical or similar tasks. This goal has been partially met as shown in the case study where the comparison of two sessions carried out by two separate subjects undertaking a similar task has enabled the identification of periods of similar and distinctly different behaviour. There is a need to build on this, however, with a common identification of areas of interest across the different sessions. This could be achieved by using the set of areas identified within one session as a reference and cross comparing with one or more other sessions to identify those AoIs which are very

similar between the sessions and those which differ significantly. The area matching methods used in the work so far (see Section 3.3) may be sufficient to carry out this matching. A common labelling of the identified matching AoIs between the different sessions would then allow automated visual analysis of patterns of attention that are similar and different between multiple subjects. Methods from geographic movement data analysis as described in Andrienko et al. [3] for eye tracking data analysis can also be tried to compare data from multiple subjects. This has not yet been implemented but has been left as future work in combination with other planned additions.

**Scalability and generalization.** All the computations applied within each time window of a particular level of the hierarchy (see Figure 5) are independent and so can be executed in parallel making this approach applicable to very long duration eye tracking datasets. To further improve the efficiency of the algorithm, the ratio  $r_{xy}$  for every edge of the MST described in Section 3.1 could be pre-computed and stored for a particular depth,  $d$ , for each eye gaze dataset. The interactive control panel (see Figure 9(c)) can be used to modify the parameters for calculation of AoIs and hence the system is readily generalizable for eye tracking datasets from different applications.

Overall the results described show that the methods developed in this work provide a superior representation of the data to that provided by 2D representations such as heat maps which provide none of the temporal information which is so important in these long-duration experiments. The simplification through the AoI identification, together with the STC display, provides direct access to both spatial and temporal variation in the subject's attention. Also, as discussed earlier, the method developed in this work to identify spatial AoIs provides superior results when compared with a traditional clustering approach for identifying AoIs such as mean shift algorithm (see Figure 2). Our user interface facilitates modifying the clustering parameters in real-time to achieve the desired clustering results and the identified AoIs are displayed in an in-context visualization where image frames from the captured stimulus can be displayed as a background to the current selected time region.

## 7 CONCLUSIONS AND FUTURE WORK

A new approach for computing temporally varying AoIs in long-duration eye-tracking data sets has been presented. The method can be parallelized and hence can be applied to very long duration eye-tracking sessions. A temporal subdivision of large data sets into overlapping time windows is employed, together with a clustering algorithm used on each time window. A hierarchical cluster merging step then identifies temporally varying AoIs by combining the cluster polygons from neighbouring time windows based on maximum connectivity in both the spatial and temporal domain. Our hierarchical approach helps the analyst to identify a desired level of temporal subdivision so that the local AoIs are not lost due to saturation of the eye-tracking data. Our aim in the future is to be able to automatically extract temporal sequences of AoIs visited by test subjects during long-duration experiments. Data mining techniques can then be used to uncover patterns of subjects' attention hidden in the long temporal sequences of AoIs which would not otherwise be easily recognized. The method described in this paper is the first step in a larger analysis pipeline being developed. This analysis pipeline will include not only the possibility to mine visual scan strategies of single subjects but also the possibility to automatically mine and compare patterns of visual behaviour among a group of test subjects participating in an eye-tracking experiment. This latter capability brings forward the interesting research challenge of how to create a common AoI alphabet for the several eye-tracking study participants. We also plan as future work to evaluate the proposed visual analysis system with the assistance of air-traffic control instructors.

## ACKNOWLEDGMENTS

We thank Åsa Svensson for providing the dataset. This work is supported by the Swedish Research Council, grant number 2013-4939 and partly by RESKILL project funded by the Swedish Transport Administration, Swedish Maritime Administration and the Swedish Air Navigation Service Provider LFV.

## REFERENCES

- [1] Tobii studio users manual. Technical Report Version 3.4.5, Tobii AB, January 2016.
- [2] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011.
- [3] G. Andrienko, N. Andrienko, M. Burch, and D. Weiskopf. Visual analytics methodology for eye movement studies. *IEEE TVCG*, 18(12):2889–2898, 2012.
- [4] N. Andrienko, G. Andrienko, G. Fuchs, S. Rinzivillo, and H.-D. Betz. Detection, tracking, and visualization of spatial event clusters for real time monitoring. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10. IEEE, 2015.
- [5] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A review of temporal data visualizations based on space-time cube operations. In *Eurographics conference on visualization*, 2014.
- [6] T. Blascheck, M. Burch, M. Raschke, and D. Weiskopf. Challenges and perspectives in big eye-movement data visual analytics. In *Big Data Visual Analytics (BDVA)*, 2015, pp. 1–8. IEEE, 2015.
- [7] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. State-of-the-art of visualization for eye tracking data. In *Proceedings of EuroVis*, vol. 2014, 2014.
- [8] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. Visualization of eye tracking data: A taxonomy and survey. In *CGF*. Wiley Online Library, 2017.
- [9] T. Blascheck, K. Kurzhals, M. Raschke, S. Strohmaier, D. Weiskopf, and T. Ertl. AoI hierarchies for visual exploration of fixation sequences. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pp. 111–118. ACM, 2016.
- [10] M. Burch, G. Andrienko, N. Andrienko, M. Höferlin, M. Raschke, and D. Weiskopf. Visual task solution strategies in tree diagrams. In *PacificVis*, 2013, pp. 169–176. IEEE, 2013.
- [11] M. Burch, F. Beck, M. Raschke, T. Blascheck, and D. Weiskopf. A dynamic graph visualization perspective on eye movement data. In *Proceedings of the Symposium on ETRA*, pp. 151–158. ACM, 2014.
- [12] M. Burch, L. Chuang, B. Fisher, A. Schmidt, and D. Weiskopf. *Eye Tracking and Visualization: Foundations, Techniques, and Applications. ETVIS 2015*. Springer, 2017.
- [13] M. Burch, A. Kull, and D. Weiskopf. AoI rivers for visualizing dynamic eye gaze frequencies. In *Computer Graphics Forum*, vol. 32, pp. 281–290. Wiley Online Library, 2013.
- [14] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.
- [15] C. E. Connor, H. E. Egeth, and S. Yantis. Visual attention: bottom-up versus top-down. *Current biology*, 14(19):R850–R852, 2004.
- [16] G. Drusch, J. Bastien, and S. Paris. Analysing eye-tracking data: From scanpaths and heatmaps to the dynamic visualisation of areas of interest. *STHESCA*, 20:205, 2014.
- [17] 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*. ACM, 2010.
- [18] A. T. Duchowski, M. M. Price, M. Meyer, and P. Orero. Aggregate gaze visualization with real-time heatmaps. In *Proceedings of the Symposium on ETRA*, pp. 13–20. ACM, 2012.
- [19] S. Fortune. Handbook of discrete and computational geometry. chap. Voronoi Diagrams and Delaunay Triangulations, pp. 377–388. CRC Press, Inc., Boca Raton, FL, USA, 1997.
- [20] J. H. Goldberg and J. I. Helfman. Scanpath clustering and aggregation. In *Proceedings of the 2010 symposium on ETRA*, pp. 227–234. ACM, 2010.
- [21] R. B. Goldstein, R. L. Woods, and E. Peli. Where people look when watching movies: Do all viewers look at the same place? *Computers in biology and medicine*, 37(7):957–964, 2007.
- [22] Y. Gu, C. Wang, R. Bixler, and S. D’Mello. Etgraph: A graph-based approach for visual analytics of eye-tracking data. *Computers & Graphics*, 62:1–14, 2017.
- [23] T. Hägerstrand. What about people in regional science. *Regional Science Association Papers*, 14:6–21, 1970.
- [24] J. Heminghus and A. T. Duchowski. icomp: a tool for scanpath visualization and comparison. In *ACM SIGGRAPH 2006 Research posters*, p. 186. ACM, 2006.
- [25] 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.
- [26] 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.
- [27] T. C. Kübler, K. Sippel, W. Fuhl, G. Schievelbein, J. Aufreiter, R. Rosenberg, W. Rosenstiel, and E. Kasneci. Analysis of eye movements with eyetrace. In *International Joint Conference on Biomedical Engineering Systems and Technologies*, pp. 458–471. Springer, 2015.
- [28] K. Kurzhals, F. Heimerl, and D. Weiskopf. ISeeCube: Visual analysis of gaze data for video. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 43–50. ACM, 2014.
- [29] K. Kurzhals, M. Hlawatsch, F. Heimerl, M. Burch, T. Ertl, and D. Weiskopf. Gaze stripes: Image-based visualization of eye tracking data. *IEEE TVCG*, 22(1):1005–1014, 2016.
- [30] K. Kurzhals, M. Hlawatsch, C. Seeger, and D. Weiskopf. Visual analytics for mobile eye tracking. *IEEE TVCG*, 23(1):301–310, 2017.
- [31] K. Kurzhals and D. Weiskopf. Space-time visual analytics of eye-tracking data for dynamic stimuli. *IEEE TVCG*, 19(12):2129–2138, 2013.
- [32] K. Kurzhals and D. Weiskopf. AoI transition trees. In *Proceedings of the 41st Graphics Interface Conference*, pp. 41–48. Canadian Information Processing Society, 2015.
- [33] X. Li, A. Çöltekin, and M.-J. Kraak. Visual exploration of eye movement data using the space-time-cube. In *International Conference on Geographic Information Science*, pp. 295–309. Springer, 2010.
- [34] L. E. Matzen, M. J. Haass, K. M. Divis, Z. Wang, and A. T. Wilson. Data visualization saliency model: A tool for evaluating abstract data visualizations. *IEEE transactions on visualization and computer graphics*, 24(1):563–573, 2018.
- [35] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson. Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, 3(1):5–24, 2011.
- [36] P. K. Muthumanickam, C. Forsell, K. Vrotsou, J. Johansson, and M. Cooper. Supporting exploration of eye tracking data: Identifying changing behaviour over long durations. *BELIV ’16*, pp. 70–77. ACM, New York, NY, USA, 2016.
- [37] L. Nachmanson, A. Nocaj, S. Bereg, L. Zhang, and A. Holroyd. Node overlap removal by growing a tree. In *International Symposium on Graph Drawing and Network Visualization*, pp. 33–43. Springer, 2016.
- [38] 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.
- [39] Y. Pinto, A. R. van der Leij, I. G. Sligte, V. A. Lamme, and H. S. Scholte. Bottom-up and top-down attention are independent. *Journal of vision*, 13(3):16–16, 2013.
- [40] C. M. Privitera and L. W. Stark. Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *IEEE Transactions on pattern analysis and machine intelligence*, 22(9):970–982, 2000.
- [41] D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pp. 71–78. ACM, 2000.
- [42] A. Santella and D. DeCarlo. Robust clustering of eye movement recordings for quantification of visual interest. In *Proceedings of the 2004 symposium on ETRA*, pp. 27–34. ACM, 2004.
- [43] T. Santini, W. Fuhl, T. C. Kübler, and E. Kasneci. Bayesian identification of fixations, saccades, and smooth pursuits. In *Proceedings of the Ninth Biennial ACM Symposium on ETRA, 2016, Charleston, SC, USA, March 14–17, 2016*, pp. 163–170, 2016.
- [44] Y. Sawahata, R. Khosla, K. Komine, N. Hiruma, T. Itou, S. Watanabe, Y. Suzuki, Y. Hara, and N. Issiki. Determining comprehension and quality of tv programs using eye-gaze tracking. *Pattern Recognition*, 41(5):1610–1626, 2008.
- [45] H. Y. Tsang, M. Tory, and C. Swindells. eSeeTrack - visualizing sequential fixation patterns. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):953–962, 2010.
- [46] K. Vrotsou, J. Johansson, and M. Cooper. Activitree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE TVCG*, 15(6):945–952, 2009.
- [47] K. Vrotsou and A. Nordman. Exploratory visual sequence mining based on pattern-growth. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018. doi: 10.1109/TVCG.2018.2848247
- [48] D. S. Wooding. Eye movements of large populations: II. deriving regions of interest, coverage, and similarity using fixation maps. *Behavior*

- Research Methods*, 34(4):518–528, 2002.
- [49] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers*, 100(1):68–86, 1971.