The Hierarchical Flow of Eye Movements*

Michael Burch
TU Eindhoven
m.burch@tue.nl

Ayush Kumar
Stony Brook University
aykumar@cs.stonybrook.edu

Klaus Mueller
Stony Brook University
mueller@cs.stonybrook.edu

ABSTRACT

Eye movements are composed of spatial and temporal aspects. Moreover, not only the eye movements of one subject are of interest, but a data analyst is more or less interested in the scanning strategies of a group of people in a condensed form. This data aggregation can provide useful insights into the visual attention over space and time leading to the detection of possible visual problems or design flaws in the presented stimulus. In this paper we present a way to visually explore the flow of eye movements, i.e., we try to bring a layered hierarchical structure into the spatio-temporal eye movements. To reach this goal, the stimulus is spatially divided into areas of interest (AOIs) and temporally or sequentially aggregated into time periods or subsequences. The weighted AOI transitions are used to model directed graph edges while the AOIs build the graph vertices. The flow of eye movements is naturally obtained by computing hierarchical layers for the AOIs while the downward edges indicate the hierarchical flow between the AOIs on the corresponding layers.

CCS CONCEPTS

- **Human-centered computing** → **Graph drawings**;

KEYWORDS

Eye tracking, Areas of interest, Hierarchical layout

ACM Reference Format:
Michael Burch, Ayush Kumar, and Klaus Mueller. 2018. The Hierarchical Flow of Eye Movements. In *ETVIS'18: 3rd Workshop on Eye Tracking and Visualization , June 14–17, 2018, Warsaw, Poland.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3205929.3205930

1 INTRODUCTION

It is oftentimes challenging to judge the general trend in eye movements, meaning the inherent flow over a stimulus in order to answer a given task. This flow highly depends on the stimulus content but also on the task itself [20] that may be composed of several subtasks or subdivided into those by the visual observers. Although the flow may be detectable for an individual observer, for a larger group it is pretty hard to see in a traditional visualization. Consequently, a more algorithmic data preprocessing, transformation, and aggregation step is required visualized as a layered node-link diagram.

Generally, visualization techniques like gaze plots [18] or scanpath visualizations [9] do not make a difference for the flow of those eye movements nor do they provide an aggregated form with the goal to derive a general movement trend in the data. These diagram types normally produce a flat temporal or sequential order and place the individual scanpaths aligned with the time axis. But the major problem comes into play if more visual scanning strategies are applied leading to some kind of branching time behavior [1].

In this paper we investigate the problem of computing the general flow of eye movements while a graph-based visualization [8] is presented that depicts this flow behavior. The eye movement data is first transformed into weighted transitions based on the AOI subdivision of a stimulus. Then the visual attention order of those AOIs in each of the scanpaths is taken into account and an average order for each AOI describes the layer on which it will be lying in the final layout. The order of the AOIs on each layer is optimized based on the aggregated and weighted transitions with the goal to reduce the number of link crossings. Finally, the links are drawn as weighted transitions between those AOIs as color coded and differently thick lines indicating major and minor flows.

The described approach is related to the standard layered, hierarchical, or Sugiyama layout [19], but for time efficiency and problem adaptation reasons we provide a version with a reduced number of layout computation steps while also taking into account the AOI order for the layering.

2 RELATED WORK

Eye movements of several people [4] can produce vast amounts of visual clutter [17] if they are drawn on top of the observed stimulus. This can either partially or totally hide the stimulus and also the other scanpaths completely. This problem is due to the restriction to the spatial arrangement of the scanpaths given by the underlying stimulus. Hence, although such gaze plot representations [18] have been used for a long time by eye tracking researchers, they cannot really reflect eye movement patterns [2] nor can they show a general trend, i.e., a flow in the data if scanpaths were recorded over longer time spans or many participants took part in an eye tracking experiment.

Also visual attention maps [5], although they do not use line-based visual encodings of the scanpaths, are problematic since they aggregate the data over time, meaning no time-varying behavior is visualized. The visual attention maps by

*The Hierarchical Flow of Eye Movements

ETVIS'18, June 14–17, 2018, Warsaw, Poland

ACM ISBN 978-1-4503-5787-6/18/06...$15.00
https://doi.org/10.1145/3205929.3205930

Burch [6] preserve the time information, but branching and merging aspects and different strengths of the transitions are difficult to see. This perceptually challenging issue is caused by using color coding instead of relying on node-link diagrams in a specific layout, like some kind of hierarchical representation [19].

Typical scanpath visualizations [9, 16] focus on depicting the individual eye movements on a linear timeline in parallel with the goal to untangle the hairball given in typical gazeplots. Although this is a good strategy, there is no aggregated way to show the general trends or flows in the data, but this has to be done visually by, e.g., inspecting the scanpath visualizations for certain patterns.

In this paper we got inspired by such a layered layout while we adapt the approach to our scenario by computing a layering first on a classing-based average AOI visiting order, modeling the weighted AOI transitions in between, and finally, modifying the AOI positions on the layers with the goal to reduce the crossing of the transition links similar to the work of Dujmovic et al. [10].

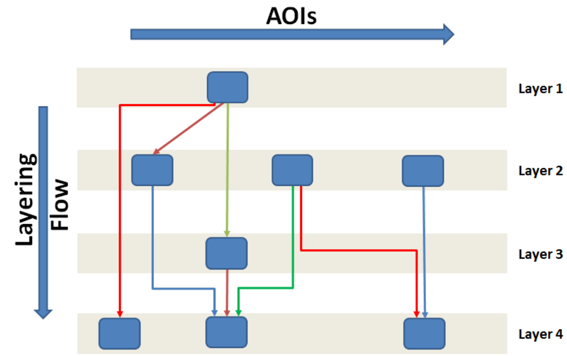


Figure 1: Mapping areas of interest (AOIs) to layers depending on their average order in scanpaths. The AOIs on each layer are permuted in order to reduce the amount of visual clutter caused by crossing links.

3 DATA TRANSFORMATION AND LAYOUT

From a set of scanpaths we compute AOIs based on several criteria, transform the fixation sequences into AOI sequences, assign each AOI an average order in the scanpaths, and finally, map those AOIs to layers and improve the positions based on weighted AOI transitions.

3.1 Eye Movements and Scanpaths

For the purpose of this research, we model eye movement data mathematically as a sequence of fixation points over space, i.e., for each participant we record

$$S_i := \left(p_{i,1}, \ldots, p_{i,n_i}\right)$$

during the eye tracking experiment. The $p_{i,j} \in \mathbb{N} \times \mathbb{N}$ express the fixation points (here in 2D space) while it may be noted that each scanpath may consist of a different number of fixation points.

3.2 AOIs and Average AOI Order

We support several ways of generating areas of interest depending on the demands of the data analysts and the tasks they are planning to solve. Those are spatial grid-based, stimulus semantics-based, or visual attention data-driven approaches. The default setting in which AOIs are automatically defined is based on a Voronoi separation taking the visual attention hot spots as cell centroids and a temporal aggregation into 10 time intervals.

By applying any of those AOI generation methods, each of the scanpaths S_i is transformed into an AOI sequence

$$A_i := \left(A_{i,1}, \ldots, A_{i,n_i}\right).$$

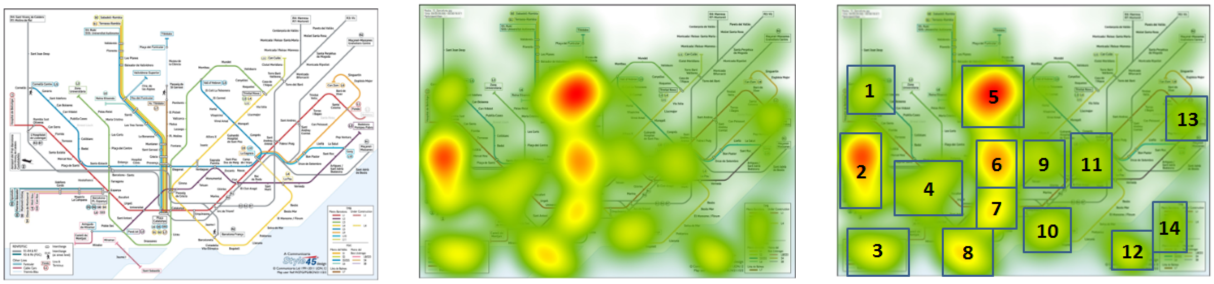
The generated AOI sequence can further be reduced by length by subdividing the time axis into equally long time intervals. We might also compute intervals by looking at which AOI the most time was spent, but by this strategy we would obtain a non-linear temporal aggregation which introduces some kind of temporal lie factor. The general goal of the temporal aggregation into time intervals is to reduce the number of AOIs occurring in each AOI sequence since we only incorporate the temporally first AOI in each time interval into the further analysis. This reduces the algorithmic runtime complexity on the one hand but also gives a more temporally coarse-grained impression of the data.

From the remaining AOI sequences A_i we compute weighted transitions pairwisely that model the number of eye movements that happened between two AOIs. The AOIs and the transitions result in a weighted directed graph that may be visualized by any layout [3] but a hierarchical-like layout is best suited for reflecting the flow of the eye movements.

Moreover, we also use the AOIs to compute an average order of each AOI in all of the sequences. To reach this goal, we first assign each AOI in each sequence a number of occurrence in the sequence, i.e., the position in the sequence. We sum up all position numbers and based on this sum and the total number of occurrences we compute the average occurrence number for each AOI globally. It may be noted that an individual AOI can occur many times in a sequence which is then assigned many occurrence numbers. However, in the end we obtain a list of AOIs L_A filled with real-valued numbers expressing the average occurrence in all of the scanpaths based on the AOI subdivision strategy and the temporal aggregation.

3.3 Layers and Layout

The average occurrence number of each AOI is used for computing the layers on which an AOI is placed. But before doing that, a classing is done on users' demand to reduce the number of layers. In the default setting the algorithm computes 5 classes based on the range of the average values. Consequently, each AOI is attached by a new number indicating the layer (the class) on which it will be placed in the final layout. It may be noted that the more classes are



(a) (b) (c)

Figure 2: The Barcelona map in which the task was to find a route from the airport to the station Vallvidrera Superior: The original stimulus (a) overplotted with the visual attention map (b) and annotated with visual attention data-driven AOIs (c).

defined, the more layers will be in the final layout. Empty layers will be removed automatically.

In the next step we recompute the order of the AOIs on each layer to reduce the crossing number of the links depicting the weighted transitions [10]. This can be done by interpreting the layers and the AOIs as a rectangular grid while the cells are traversed horizontally until the overlap is minimized which can be solved by a heuristics of the MinLA problem [11]. The drawing of the transition links can either be done as straight lines between the centers of the corresponding AOI boxes or as orthogonal links avoiding overlaps with AOI boxes. Figure 1 shows an illustration of how the AOIs might be placed while orthogonal links are used for depicting the weighted AOI transitions.

4 HIERARCHICAL FLOW VISUALIZATION

The visualization provides a quick overview about the flow of eye movements by showing a node-link diagram in which the general flow starts at the top and is heading to the bottom.

4.1 Design Decisions

We designed an interactive visualization focusing on several aspects. The intention of our work is to get an overall impression about the flow of eye movements from a larger group of people. We could do this by aggregating the visual scanning strategies like in edge bundling approaches [12, 13, 15] but this causes the problem of overplotting the static stimulus. This idea is not novel and as a drawback it would suffer from visual clutter [17], for example, if cross checking or back and forth visual scanning strategies are applied [7]. The flow is not easily derivable if it is depicted on top of the stimulus while finally, also the stimulus would be difficult to interpret.

To reach our goal we support several AOI generation models as described in Section 3.2 and also temporal aggregation into time intervals for reducing the amount of data spatially and temporally. The averaging of AOI occurrences from many AOI sequences can be a useful strategy to visually explore the hierarchical flow of eye movements. Hence, averaging makes sense in order to derive a layering of the AOIs first which is one of the differences to the hierarchical layout by Sugiyama, i.e., the layering is already given in our approach in contrast to the Sugiyama technique in which it is explicitly computed. The AOI transition data is taken into account as a very last step to modify the final layout, i.e., to reduce link crossings. If we filter by transition weights first and then determine if the resulting graph actually represents a hierarchy, then this hierarchy would not follow the reading direction in general, hence it would lead to misinterpretations of the data.

4.2 Layer-Based Node-Link Diagram

The AOIs are modeled as vertices of the graph and are visualized as rectangular boxes. Those boxes are placed on vertically stacked layers, while each layer indicates an average occurrence class an AOI falls in. The AOIs on each layer are equidistantly placed while also the vertical distance between the layers follows an equidistant mapping. The layers start from top to bottom with increasing class numbers leading to a top-to-bottom flow reading direction (see Figure 1).

The weighted transitions are modeled as directed graph edges and are placed between the nodes (AOIs) of the graph. The weights are mapped to a user-selected color coding and are additionally indicated by link thicknesses. The directions of the transitions are given by arrow heads, i.e., they are required since there might also be upwardly heading links, not following the general flow trend. In the examples in this paper we do not need arrow heads since we removed non-downward pointing transition links since they do not reflect the hierarchical flow.

5 APPLICATION EXAMPLE

To illustrate the usefulness of our technique we applied it to eye movement data that was formerly recorded [14]. The data consists of 40 scanpaths for each public transport map. To show the usefulness we picked the map of Barcelona (Figure 2 (a)). The task for each participant was to find a route from the airport to the station Vallvidrera Superior. The metro map in Figure 2 (b) is overplotted by a visual attention map showing the hotspots of visual attention aggregated over time and over all 40 participants. Although the distribution of visual attention frequency is depicted we cannot derive insights about time-varying scanning behavior nor can we see if there is a general flow of the eye movements while the participants try to answer the route finding task.

For this reason we select a manual visual attention data-driven AOI subdivision (Figure 2 (c) annotated with AOIs and numbers). We subdivide each of the 40 scanpaths into time periods of 500 ms, generate AOI sequences, and finally remove those fixations that did not fall into any of the selected AOIs. Average occurrences, weighted transitions, and AOI classes are computed.

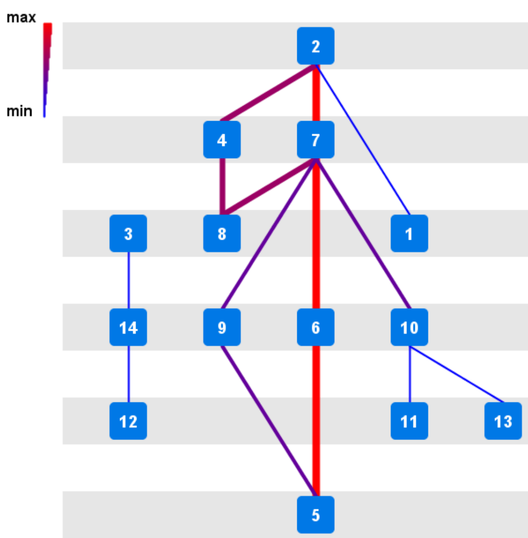


Figure 3: Hierarchical flow of eye movements computed for the route finding task in the Barcelona metro map.

Figure 3 shows an example of the final result of the hierarchical flow generation algorithm. The classing was set to produce exactly 6 layers but can be changed interactively. We could show all transitions in this diagram, but since we are only interested in the hierarchical flow and the most important ones, we filtered out all the upward edges and also those that point between AOIs located on the same layer.

In Figure 3 we can see for example, that AOI 2 is placed on top, reflecting that most of the people start their visual inspection there. This is a normal strategy since the starting point (the airport) is located there. Then the algorithm says that there is one major solution with a sequence of AOIs 7, 6, and 5 (the end station). Looking at the stimulus we are wondering what happened with AOIs 4 and 8. They are lying on the solution route but are placed in a branch by the algorithm. This means they are possible candidates but most of the participants directly moved their eyes from 2 to 7 and not from 2 to 4 or 8 first. Maybe, the peripheral vision is to blame here, i.e., the observer sees the next step already without focusing on intermediate steps.

There is another stand-alone branch indicating that AOIs 3, 14, and 12 are connected. Looking at the stimulus again, we see that they belong to the map legends, but they are only visited by a few participants and they are placed in the bottom layers, meaning they have been inspected rather in the end of the visual scanning. Maybe some people had problems with the map understanding and tried to understand the legends before proceeding with the task solution. There are many more insights in this static diagram but if we are applying the interactions and change parameters, we can even find more strategies and outliers.

6 DISCUSSION AND LIMITATIONS

There are several limitations worth discussing, e.g., taking into account data transformation, data noise, data aggregation, or data space.

- **Data transformation:** Runtime complexities are not that high, i.e., the tool can be interactively used. If the number of scanpaths, the number of fixations in the scanpaths, or the number of AOIs grows, this will cause longer run times.
- **Data noise:** If there is no general trend or several subtrends in eye movement data, then the algorithm would compute average AOI occurrences that are nearly on the same level, i.e., they would be placed on the same or neighbored layer.
- **Data aggregation:** Individual participants cannot be detected anymore in the hierarchical diagram. Those have to be found by further interaction techniques using another representation to select individual participants.
- **Data space:** The spatial arrangement of the visual attention data is changed since it is abstracted into graph data. Overlaying the stimulus with the weighted AOI transition graph would make the flow difficult to be observed due to visual clutter problems.

7 CONCLUSION AND FUTURE WORK

In this paper we presented the hierarchical flow of eye movements. The algorithm is able to generate a top-to-bottom flow diagram showing the general trend in eye movements. The scanpaths are transformed into AOI sequences while the AOI occurrences are averaged over all sequences. Weighted transitions between the AOIs are computed, i.e., AOIs model the vertices of the graph and the transitions the weighted edges. The classing of the average values decides about the layers the AOIs are placed while they are permuted in order to achieve a clutter-reduced hierarchical layout. We showed the usefulness of our approach by applying it to formerly recorded eye movement data from a public transport map experiment. For future work we plan to add more interaction techniques, while also non-static stimuli might be of interest for detecting a flow in the visual scanning data.

8 ACKNOWLEDGMENTS

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Consilience Creative Program (IITP-2017-R0346-16-1007) supervised by the IITP (Institute for Information & Communications Technology Promotion).

REFERENCES

[1] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. 2011. *Visualization of Time-Oriented Data*. Springer.

[2] Gennady Andrienko, Natalia Andrienko, Michael Burch, and Daniel Weiskopf. 2012. Visual Analytics Methodology for Eye Movement Studies. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2889–2898.

[3] Tanja Blascheck, Michael Burch, Tobias Meisel, Tobias Schneider, and Safak Mumin. 2018. Exploring Eye Movements with Node-Link Graph Layouts. In *Proceedings of Workshop on Eye Movements for Spatial Research (ET4S)*.

[4] Tanja Blascheck, Michael Burch, Michael Raschke, and Daniel Weiskopf. 2015. Challenges and Perspectives in Big Eye-Movement Data Visual Analytics. In *Proceedings of the 1st International Symposium on Big Data Visual Analytics*. 17–24.

[5] Agnieszka Bojko. 2009. Informative or Misleading? Heatmaps Deconstructed. In *Human-Computer Interaction – INTERACT*. Springer, 30–39.

[6] Michael Burch. 2016. Time-Preserving Visual Attention Maps. In *Proceedings of Conference on Intelligent Decision Technologies*. 273–283.

[7] Michael Burch, Gennady L. Andrienko, Natalia V. Andrienko, Markus Höferlin, Michael Raschke, and Daniel Weiskopf. 2013. Visual Task Solution Strategies in Tree Diagrams. In *Proceedings of IEEE Pacific Visualization Symposium*. 169–176.

[8] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. 1999. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall.

[9] Andrew T. Duchowski, Jason Driver, Sheriff Jolaoso, William Tan, Beverly N. Ramey, and Ami Robbins. 2010. Scanpath comparison revisited. In *Proceedings of the Symposium on Eye-Tracking Research & Applications, ETRA*. 219–226.

[10] Vida Dujmovic, Henning Fernau, and Michael Kaufmann. 2008. Fixed parameter algorithms for one-sided crossing minimization revisited. *Journal of Discrete Algorithms* 6, 2 (2008), 313–323.

[11] M. R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

[12] Danny Holten. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transaction on Visualization and Computer Graphics* 12, 5 (2006), 741–748.

[13] Danny Holten and Jarke J. van Wijk. 2009. Force-Directed Edge Bundling for Graph Visualization. *Computer Graphics Forum* 28, 3 (2009), 983–990.

[14] Rudolf Netzel, Bettina Ohlhausen, Kuno Kurzhals, Robin Woods, Michael Burch, and Daniel Weiskopf. 2017. User performance and reading strategies for metro maps: An eye tracking study. *Spatial Cognition & Computation* 17, 1–2 (2017), 39–64.

[15] Vsevolod Peysakhovich, Christophe Hurter, and Alexandru Telea. 2015. Attribute-driven edge bundling for general graphs with applications in trail analysis. In *IEEE Pacific Visualization Symposium*. 39–46.

[16] Michael Raschke, Dominik Herr, Tanja Blascheck, Thomas Ertl, Michael Burch, Sven Willmann, and Michael Schrauf. 2014. A visual approach for scan path comparison. In *Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA*. 339–346.

[17] Ruth Rosenholtz, Yuanzhen Li, Jonathan Mansfield, and Zhenlan Jin. 2005. Feature Congestion: A Measure of Display Clutter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 761–770.

[18] Leonard F. Scinto, Ramakrishna Pillalamarri, and Robert Karsh. 1986. Cognitive Strategies for Visual Search. *Acta Psychologica* 62, 3 (1986), 263–292.

[19] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. 1981. Methods for Visual Understanding of Hierarchical System Structures. *IEEE Transactions on Systems, Man, and Cybernetics* 11, 2 (1981), 109–125.

[20] A. L. Yarbus. 1967. *Eye Movements and Vision (Translated from Russian by Basil Haigh. Original Russian edition published in Moscow in 1965.)*. New York: Plenum Press.