Capturing and Explaining Trajectory Singularities using Composite Signal Neural Networks

Hippolyte Dubois*, Patrick Le Callet*, Michael Hornberger†, Hugo J. Spiers‡, Antoine Coutrot*
*Université de Nantes, CNRS, LS2N, F-44000 Nantes, France
{hippolyte.dubois, patrick.lecallet, antoine.coutrot}@univ-nantes.fr
† Norwich Medical School, University of East Anglia, Norwich, United Kingdom
‡ Institute of Behavioural Neuroscience, Department of Experimental Psychology,
Division of Psychology and Language Sciences,
University College London, London, United Kingdom

***Abstract*—Spatial trajectories are ubiquitous and complex signals. Their analysis is crucial in many research fields, from urban planning to neuroscience. Several approaches have been proposed to cluster trajectories. They rely on hand-crafted features, which struggle to capture the spatio-temporal complexity of the signal, or on Artificial Neural Networks (ANNs) which can be more efficient but less interpretable. In this paper we present a novel ANN architecture designed to capture the spatio-temporal patterns characteristic of a set of trajectories, while taking into account the demographics of the navigators. Hence, our model extracts markers linked to both behaviour and demographics. We propose a composite signal analyser (CompSNN) combining three simple ANN modules. Each of these modules uses different signal representations of the trajectory while remaining interpretable. Our CompSNN performs significantly better than its modules taken in isolation and allows to visualise which parts of the signal were most useful to discriminate the trajectories.**

***Index Terms*—graph signal processing, neural network, cnn, gcnn, explainability, trajectory, pattern analysis**

I. Introduction

Understanding and modelling the basic laws governing human spatial navigation is crucial is many fields such as urban planning [1], traffic forecasting [2], activity understanding [3], ecology [4], behavioural and clinical neuroscience [5], see [6] for a review. The rapid spread of GPS devices, sensor network, satellite and wireless communication technology, enables the tracking of all kinds of moving objects all over the world. This results in an increasing number of moving object trajectory data to be collected and stored in large-scale databases [7]. Spatial trajectories are rich signals that are not straightforward to handle, notably because they are determined by a complex interaction between the shape of the environment (walls, obstacles: bottom-up determinants) and the navigator's decisions (task at hand, cognitive state: top-down determinants). To mine these complex spatial patterns, numerous approaches have been proposed in the literature. The most popular approach is trajectory clustering, which aims at discovering groups of similar trajectories in an unsupervised, semi-supervised, or supervised manner. Trajectory clustering typically quantifies trajectories with a set of features (e.g. coordinates, angle, speed, curvature…) and then apply clustering algorithms such as **K-means** or Density-Based Spatial Clustering of Applications with Noise (**DBSCAN**) [8]. Another approach is to use trajectory similarity measures such as dynamic time warping (**DTW**), edit distance on real sequence (**EDR**), longest common subsequences (**LCSS**) and then apply a clustering algorithm in the space defined by these metrics [9].

These classical data mining techniques have the advantage of being highly interpretable, but they are quite limited for several reasons. First, trajectories intrinsically have spatial and temporal properties, which conventional approaches often can not handle at the same time. Second, trajectories are strongly self-correlated (position at time t is highly dependent on position at time $t-1$). This violates the assumption of independence of samples, on which many conventional clustering approaches depend. To overcome these limitations, artificial neural networks (**ANN**) have been proposed with success, as they can learn efficient representations directly from the raw data, without the need to hand-craft the features [10]. ANN can also jointly capture spatial patterns (e.g. with convolutional neural networks, **CNN**) and temporal patterns (e.g. with recurrent neural networks, **RNN**). The main limitations of ANNs for trajectory analysis is that while being efficient in a certain context, they poorly generalize to other situations. Furthermore, while ANN explainability has recently been in the spotlight in some research fields (particularly in medecine [11]), this has less been the case in trajectory analysis, where they mostly remain black-boxes.

II. Contributions

In this study we propose a method to capture the spatio-temporal patterns characteristic of the trajectories followed by a given population navigating in a constrained environment. We aim to associate these patterns to the demographics of the navigators, to extract clusters homogeneous in both *behaviour* and *demographics*. To extract meaningful and interpretable trajectory markers, we propose a composite signal analyser (**CompSNN**). We hypothesize that, by combining simple models capturing different aspects of the data, we can obtain good analysis performance, while maintaining a high level of explainability. Our analyser uses three dependencies between the samples of our data:

- Temporal dependency using time-series representation, with a CNN
- Spatial dependency using graph representation, with a Multi Layer Perceptron (**MLP**)
- Spatial dependency using graph spectrum representation, with a Graph convolutional Neural Network (**GCNN**)

The output of those networks is then concatenated, and fed into a MLP, which combines temporal and spatial signal analysis, hence the Composite Signal. This MLP is trained to predict the demographic from the behaviour. Our code is available on https://gitlab.univ-nantes.fr/E173825Q/compsnn-eusipco2020.

III. Architecture of the CompSNN model

The CompSNN is an aggregate of three different ANNs: a MLP, a GCNN, and a CNN. In this section, we describe each of these components.

A. Components

a) MLP: The first component is a MLP taking the signal defined on the graph $\mathcal{G}(\mathbf{N}, \mathbf{E})$ nodes as input. It is agnostic of the graph's structure, so it's important that the graph modeling the data is designed to capture the singularity of the signal. We fully describe the graph layout section (see Fig. V).

The MLP is composed of a first linear layer, with $|\mathbf{N}| * \mathbf{8}$ inputs, 32 hidden nodes, and 16 outputs, with a ReLU non-linearity activation in the middle. It processes signal defined on the nodes of the graph, each node holding values in $\mathbb{R}^8$, namely:

- the mean of the speed s,
- the mean of the acceleration ∇s,
- the mean of the direction $\theta = atan2(x, y)$,
- the mean of the curvature $\nabla\theta$,
- the mean of the entropy $\mathbf{s}(\nabla\theta)$,
- the mean of the variance $\sigma^2(\nabla\theta)$,
- if the navigator exited the node then came back,
- the number of time it was on the node.

b) GraphCNN: The second component is a Graph convolutional Neural Network (see Fig. 2a), that learns a set of j polynomial filters H. Each of those filters is the output of a MLP which takes the eigenvalues of the graph $\mathcal{G}$ as input [12]. The input signal, that is the number of time the navigator was on each node, is then processed as follows:

$$\begin{aligned} h_k &= MLP_{h_k}(\lambda) \\ g(h) &= \sum_{k=0}^{K} h_k \Lambda^k \\ f_h(s) &= g(h) \times \hat{s} \\ f(s) &= \sum_{h \in H} \oplus f_h(s) \end{aligned} \tag{1}$$

with $\times$ the piecewise product of two vectors, $\Lambda = diag(\lambda)$, λ_i the i-th eigenvalue of the graph's Laplacian, and $\sum \oplus$ the concatenation of several vectors.

As product in spectral domain is equivalent to convolution in spatial domain, this is essentially a convolutional layer. One could argue that this is not rigorously a GCNN as the filters are not localized, but, because we do not go back to the spatial domain and we do not stack convolutional layers, we do not think it is necessary to implement localization, as it makes the model a lot more computationally expensive.

The output of the convolutional layer is then fed into a linear layer with $k|\mathbf{N}|$ inputs and 16 outputs.

c) CNN: The third component is a straight-forward 1D CNN (see Fig. 2b), with a convolutional layers of 16 features, as well as an attention layer with a 1-dimensional output, followed up with a linear layer with 16 outputs. The output of the feature-convolutional layer is multiplied by the weight outputed by the attention-convolutional layer, both of those layers providing knowledge about the data structure. Because there is only one feature-extracting layer, the model remains explainable.

d) Aggregator: All those components feed into a MLP with $16 + 16 + 16 = 48$ inputs and $|u|$ outputs.

B. Loss Function

The aggregator's output is evaluated by computing the log of the probability density function (**PDF**) of the distribution $\mathcal{N}(u, \epsilon)$. The SGD optimizer then tries to minimize the opposite of the log of the normalized $PDF(x)$.

$$loss(x, u) = -\log\left(PDF_{\mathcal{N}(u,\epsilon)}(x) - PDF_{\mathcal{N}(u,\epsilon)}(u)\right) \tag{2}$$

$\mathcal{N}(u, \epsilon)$ is the gaussian distribution centered around the N-dimensional point u sampled from the demographic space, with ϵ the standard deviation of the distribution.

We normalize the PDF because in order to compare models with different ϵ parameters, and to allow the model to know when it is right, as the maximum of the PDF is not 1. This loss function is used because, as the data is noisy, it is important to include a way to mitigate that noise when training, for example by providing the target-space structure by setting the ϵ parameter to be the variance of the demographic domain.

IV. Trajectory dataset

We use trajectories recorded in the Sea Hero Quest (**SHQ**) project [13]. SHQ is a mobile video game designed to quantify the spatial ability of players. It collected data from over 4 million people worldwide. Here we use trajectories recorded during "checkpoint levels", where players were initially presented with a map indicating the starting point and the location of several checkpoints to find in a set order. Then the map disappeared and players had to navigate their way to the checkpoints. Their trajectories T was recorded at 2 Hz, along with demographic information U known to interact with spatial ability (age, gender, education, home environment, sleep duration, commute duration, handedness and self-assessment of spatial ability). The performance at SHQ has been shown to be highly correlated to the performance at a similar spatial task in the real world [14]. Previous analyses of this dataset showed that spatial ability was indeed modulated by the player's demographics such as age and gender [13]. However these

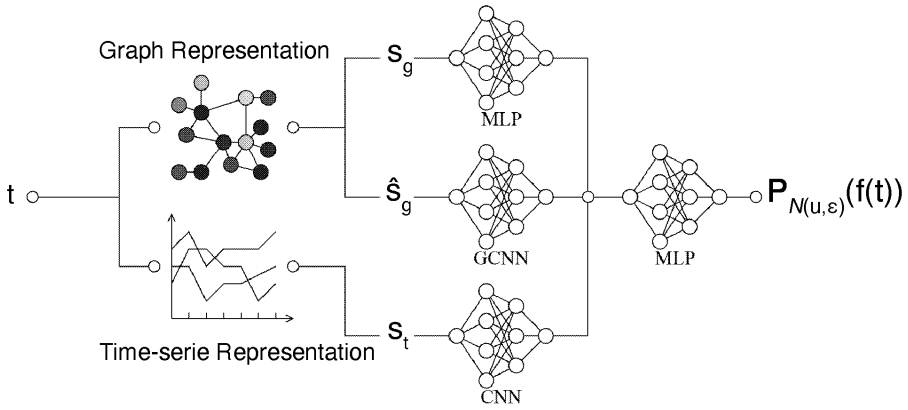


Fig. 1. Architecture of the CompSNN model. Trajectory **t** is represented by 1- a signal on a graph s_g, 2- its Graph Fourier Transform $\hat{s}_g$ and 3- time-series of trajectory features. These representations are respectively fed to a MLP, a GCNN and a CNN. The outputs of these modules are then aggregated by a MLP trained to predict the demographics from the signal extracted from the behaviour.

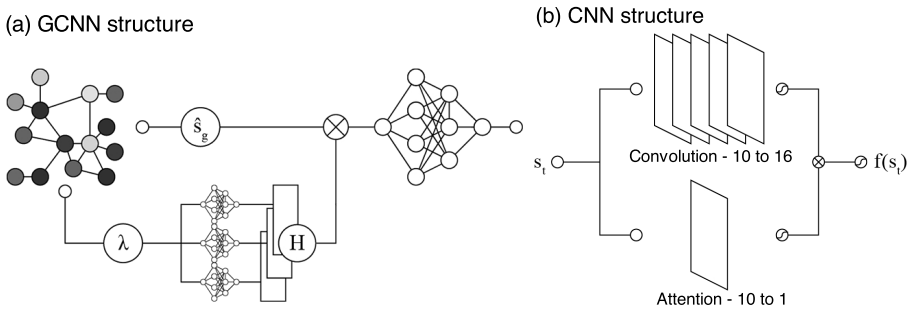
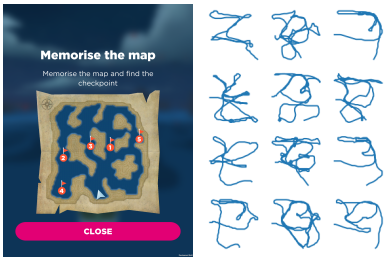


Fig. 2. Structures of the GCNN and CNN components. (a) GCNN structure, with H the filterbank, and λ the eigenvalues of $\mathcal{G}$. (b) CNN structure. The output of each layer is constrained by a Sigmoid function.

results were mostly based on the length of the trajectories, which can not fully capture the spatial strategies used by the different demographic groups. For instance, two trajectories can have the same length while having completely different shapes.

Each trajectory t of the set T (see Fig. 3b) is composed of a x and a y component, captured N times, with N being heterogeneous among T. Therefore we have $t \in \mathbb{R}^{2\times N}$.



(a) (b)

Fig. 3. The Sea Hero Quest Dataset - (a) Map of level 48. Players had to memorize the location of the 4 checkpoints and navigate to them in a set order. (b) Twelve random resulting trajectories (out of 39,289).

We can increase the dimensionality of t by computing at each timestamp n:

- the speed s,
- the acceleration ∇s,
- the x derivative ∇x,
- the y derivative ∇y,
- the direction $\theta = atan2(x, y)$,
- the curvature $\nabla\theta$,
- as well as the local entropy $\mathbf{s}(\nabla\theta)$
- and the local variance $\sigma^2(\nabla\theta)$.

We therefore have $t \in \mathbb{R}^{10\times N}$

Each user is defined by its demographic vector $u \in [0, 1]^8$. Categorical values are arbitrarily assigned values in the $[0, 1]$ interval, and ordinal values are ordered from 0 to 1 with a regular spacing.

The CompSNN defined earlier will try to learn a function $f(t|\lambda)$, with λ the parameters of the model, so that λ maximizes the probability of the output of $f(t|\lambda)$ given the distribution centered around the user's demographic vector u with covariance matrix ϵ,

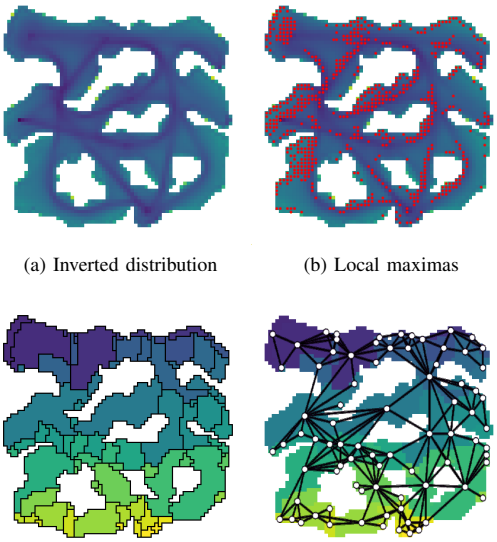
$$\max_{\lambda} \mathbb{P}_{\mathcal{N}(u,\epsilon)}(f(t|\lambda)) \tag{3}$$

with ϵ a meta-parameter used to allow the network to fail. ϵ is here set to $\sigma(U)$.

V. Defining the Graph

It is important that the structure of the graph used to quantize the input signal is fitted to our task. To capture the singularity of the signal, while maintaining the number of nodes low to avoid redundancy and reduce the complexity of the model, the graph is defined by segmenting the heatmap of the level, with the inverse density probability as a weight. That way, areas that are visited by most players are aggregated, and areas that are more significant to identifying variance, that is area that are visited only by the few, are less packed.

To do so, we use a watershed algorithm [15], with the local maximas (see Fig. 4b) of the original probability density $\mathcal{D}_d$ in the $\{x, y\}$ plane as centers, with d the level. The watershed algorithms then propagates from those centers using the inverse $\frac{1}{\mathcal{D}_d+1}$ density (see Fig. 4a). Each segment of the map (see Fig. 4c) is associated to a node of the graph $\mathcal{G}(\mathbf{N}, \mathbf{E})$. The edges are computed from the trajectories, so that $\exists \mathbf{E}_{ij}$ if at least one player went from the area associated to $\mathbf{N}_i$ to the one associated to $\mathbf{N}_j$ (sees Fig. 4d).



(a) Inverted distribution (b) Local maximas

(c) Segmented map (d) Generated graph

Fig. 4. Segmentation of the level 48 map using the watershed algorithm

The graph definition could be enhanced by using a higher dimensional distribution map, for exemple in $\{x, y, atan2(x, y)\}$. The trade-off is the increase in dimensionality of the graph signal as the number of nodes increases.

VI. Evaluation

To demonstrate the added value of using three different modules to capture different aspects of the signal, we will compare the results of the components on their own to the ones of the composite model.

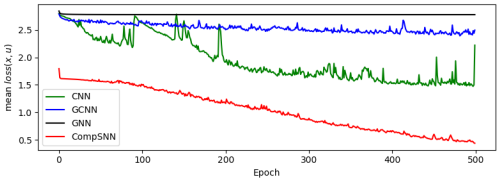
A. Data

To run the experiment, we used a subset of the 39,289 trajectories sampled from SHQ level 48 with full demographic information. We used 1000 trajectories as it is enough to show how the model performs. We used trajectories from level 48 as it is a complex level with a challenging topology, likely to elicit a larger variance in behavior than simpler levels.

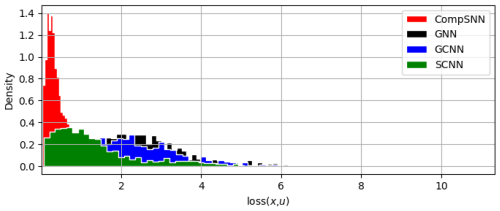
B. Structure of the SingleNNs

To compare the performance of the CompSNN to the individual modules, we train SingleNNs, which are models with a module (graph signal MLP, GCNN or CNN) feeding into a linear model with 16 inputs and $|u|$ outputs, evaluated with the same loss as the CompSNN. The weights are not shared between the CompSNN modules and the SingleNNs.

C. Comparison



(a) Mean loss per model over epochs.



(b) Loss histogram at best epoch

Model	CNN	GCNN	GNN	CompSNN
Mean	1.47	2.39	2.78	0.44
CI	[1.40;1.54]	[2.32;2.46]	[2.71;2.84]	[0.42;0.45]

(c) Mean and 95% Confidence Interval at the best epoch

Fig. 5. Comparison of the scores for each model

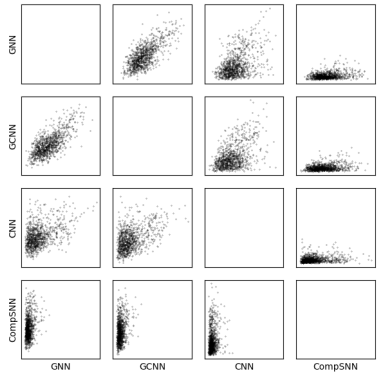


Fig. 6. Correlations between CompSNN and SingleNNs scores.

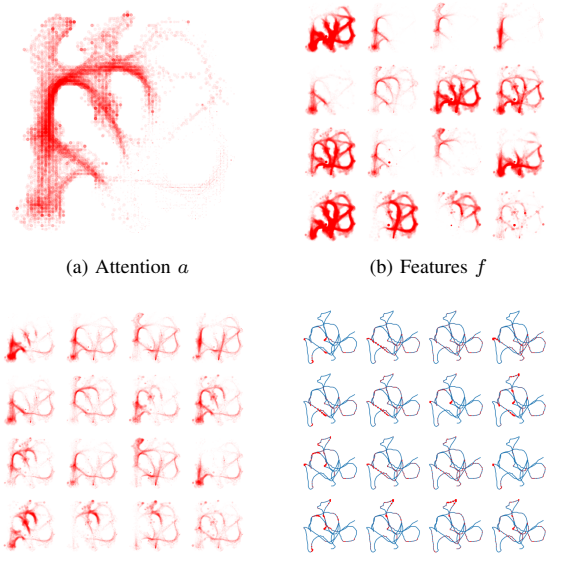
By comparing the models' performances, we can see, that, while keeping the same *deepness*, the CompSNN performs

significantly better (see Fig. 5). By providing other representations of the signal, making it *wider*, we can improve the efficiency of our system, while keeping it *shorter*. As Fig. 6 shows us, there is a correlation between the score of samples across SingleNNs, even though the CNN performs better. This suggests that a sample's *hardness* is not correlated to its signal representation, there is not a representation that would be more fitted for some samples but not others, but rather that the information can be found by aggregating different representations.

VII. Where the information lies

Because we keep the system short, we can use the learned weights to understand our data.

As it is the most visual, we will here explore the weights of the CNN component of the CompSNN (see Fig. 7). It has two components, a convolutional 10-to-16 features layer, and a 10-to-1 attention layer. We can visualize both those outputs, or the $feature \times attention$ (noted $f \times a$) output. Fig. 7a shows a visualisation of where the attention layer activates, and Fig. 7b shows how and where each feature from the feature layer is activated. Finally, Fig. 7c shows the output of the convolutional part of the CNN. As we can see, while the attention layer's output seems to be spatially correlated, not all features *look* at the same locations, and they seem to identify specific behaviours, as we can see in Fig. 7d, where some turns and straight lines are identified to be important by the model.



(a) Attention a (b) Features f

(c) $a \times f$ (d) $a \times f$ for a single trajectory

Fig. 7. Plots of the output of the convolution. The size of the dot represents the activation of the output at this point. Transparency represents the density of activation at this point.

VIII. Conclusion

In this paper, we proposed a system able to capture spatio-temporal markers of trajectories followed by navigators with specific demographic information. By combining simple explainable models capturing different dimension of the signal, we were able to build an explainable model performing significantly better than its modules taken in isolation. Future research could investigate the explainability of the GCNN module, implementing it using Koenecker delta localization, and enriching the model of other signal representations to make it more performant and give more insight on how the behavior relates to the demographics. It would also be interesting to study how the definition of the graph and of its signal influence the performances and explainability of the model. Beyond spatial navigation trajectories, this method could also be applied to other types of time series, such as eye-tracking datasets or other trajectory-like signals.

Acknowledgment

This project was partially funded by the *RFI Atlantic 2020* and *RFI Ouest Industrie Creative* programs of the French region Pays de la Loire

References

[1] B. Tang, M. L. Yiu, K. Mouratidis, and K. Wang, "Efficient Motif Discovery in Spatial Trajectories Using Discrete Fréchet Distance," 2017.

[2] P. Besse, B. Guillouet, J.-M. Loubes, and F. Royer, "Review & Perspective for Distance Based Clustering of Vehicle Trajectories," *IEEE Transactions on Intelligent Transportation Systems*, May 2016.

[3] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2016.

[4] I. Ardakani, K. Hashimoto, and K. Yoda, "Understanding Animal Behavior Using Their Trajectories," in *Distributed, Ambient and Pervasive Interactions: Technologies and Contexts*, N. Streitz and S. Konomi, Eds. Springer International Publishing, 2018.

[5] G. Coughlan, J. Laczó, J. Hort, A.-M. Minihane, and M. Hornberger, "Spatial navigation deficits — overlooked cognitive marker for preclinical alzheimer disease?" *Nature Reviews Neurology*, Jul. 2018. [Online]. Available: https://doi.org/10.1038/s41582-018-0031-x

[6] J. D. Mazimpaka and S. Timpf, "Trajectory data mining: A review of methods and applications," *Journal of Spatial Information Science*, 2016.

[7] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *The VLDB Journal*, Jan. 2020.

[8] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, "A review of moving object trajectory clustering algorithms," *Artif Intell Rev*, Jan. 2017.

[9] J. Bian, D. Tian, Y. Tang, and D. Tao, "A survey on trajectory clustering analysis," *arXiv:1802.06971 [cs]*, Feb. 2018.

[10] S. Wang, J. Cao, and P. S. Yu, "Deep Learning for Spatio-Temporal Data Mining: A Survey," Jun. 2019.

[11] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller, "Causability and explainability of artificial intelligence in medicine," *WIREs Data Mining and Knowledge Discovery*, Apr. 2019.

[12] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains," *IEEE Signal Process. Mag.*, May 2013.

[13] A. Coutrot, R. Silva, E. Manley, W. de Cothi, S. Sami, V. D. Bohbot, J. M. Wiener, C. Hölscher, R. C. Dalton, M. Hornberger, and H. J. Spiers, "Global Determinants of Navigation Ability," *Current Biology*, Sep. 2018.

[14] A. Coutrot, S. Schmidt, L. Coutrot, J. Pittman, L. Hong, J. M. Wiener, C. Hölscher, R. C. Dalton, M. Hornberger, and H. J. Spiers, "Virtual navigation tested on a mobile app is predictive of real-world wayfinding navigation performance," Mar. 2019.

[15] R. Mahmoudi and M. Akil, "Analyses of the Watershed Transform," 2011.