# SPHERICAL CLUSTERING OF USERS NAVIGATING 360° CONTENT

*Silvia Rossi*[⋆] *Francesca De Simone*[†] *Pascal Frossard*[‡] *Laura Toni*[⋆]

[⋆] Department of Electronic & Electrical Engineering, UCL, London (UK)
[†] DIS, Centrum Wiskunde & Informatica, The Netherlands
[‡] LTS4, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
E-mails: {s.rossi, l.toni}@ucl.ac.uk, F.De.Simone@cwi.nl, pascal.frossard@epfl.ch

## ABSTRACT

In Virtual Reality (VR) applications, understanding how users explore the omnidirectional content is important to optimize content creation, to develop user-centric services, or even to detect disorders in medical applications. Clustering users based on their common navigation patterns is a first direction to understand users behavior. However, classical clustering techniques fail in identifying this common paths, since they are usually focused on minimizing a simple distance metric. In this paper, we argue that minimizing the distance metric does not necessarily guarantee to identify users that experience similar navigation path in the VR domain. Therefore, we propose a graph-based method to identify clusters of users who are attending the same portion of the spherical content over time. The proposed solution takes into account the spherical geometry of the content and aims at clustering users based on the actual overlap of displayed content among users. Our method is tested on real VR user navigation patterns. Results show that our solution leads to clusters in which at least 85% of the content displayed by one user is shared among the other users belonging to the same cluster.

***Index Terms*—** Virtual Reality, 360° video, user behaviour analysis, data clustering

## 1. INTRODUCTION

Virtual Reality (VR) systems are expected to become wide spread in the near future, with applications spanning a variety of fields, from entertainment to e-healthcare. These systems involve omnidirectional ( *i.e.*, 360°) videos, which are visual signals defined on a virtual sphere depicting the 360° surrounding scene. The viewer is virtually positioned at the centre of this sphere and can navigate the scene with three Degrees-Of-Freedoms (3-DOF), *i.e.*, yaw, pitch and roll. The navigation is experienced by the user rotating his head and changing his viewing direction. This interactive navigation is typically enabled by a head-mounted display (HMD), which renders at each instant in time only the portion of the spherical content attended by the user, *i.e.*, *viewport*.

Understanding how users explore the VR content is important in order to optimize content creation [1] and distribution [2–6], develop user-centric services [7, 8], and even for medical applications that use VR to study psychiatric disorders [9]. In the last few years, many studies have appeared collecting and analysing the navigation patterns of users watching VR content [6, 8, 10–16]. Most studies build content-dependent *saliency maps* as main outcome of their analysis. The saliency map computes the most probable region of the sphere attended by the viewers, based on their head or eye movements [6, 10, 17–19]. Some studies also provide additional quantitative analysis based on metrics, such as the average angular velocity, frequency of fixation, and mean exploration angles [8, 13]. However, none of these studies provide a quantitative metric to evaluate common patterns. On the contrary, performing a clustering of navigation trajectories can show how many groups of users consistently share a similar viewport over time. The evaluation of common portion (*i.e.*, overlapped viewport) of 360° content among users could be a key-metric to evaluate users' behaviour. This information might be useful in order to improve the accuracy and robustness of algorithms predicting users navigation paths. A proper clustering could also be useful to refine user-centric distribution strategies, where for instance different groups of users might be served with higher quality content in different portions of the sphere that will be more likely displayed by the viewers. In this context, the main goal of this paper is to propose a novel clustering strategy able to detect meaningful clusters on the spherical domain. We consider as meaningful cluster a set of users *attending the same portion of spherical content*. *i.e.*, a set of users with substantial overlap between viewports. The main motivation is that a significant common overlap needs to be guaranteed for clustering methods to be used for prediction purposes or for implementing accurate user-based delivery strategies.

To the best of our knowledge, studies identifying clusters for omnidirectional content delivery have appeared only recently [20, 21]. User clustering is employed to identify the number of Region of Interests (RoIs) over time and to perform long-term trajectory prediction. In [20], the viewing directions of each user at each instant in time, *i.e.*, viewport centers, are considered as points on the equirectangular planar. These are then clustered based on Euclidean distance, neglecting the actual spherical geometry. Conversely, in [21] each user navigation pattern is modelled as independent trajectories in roll, pitch, and yaw angles, and spectral clustering is then applied. While it is efficient in discovering general trends of users' navigation, this clustering methodology is not focused on identifying clusters that are consistent in terms of overlap between viewports displayed by different users. This means that users in the same cluster do not necessarily consume the same portion of content. In other words, the identified clusters are not necessarily meaningful in the perspective of studying 360° navigation patterns.

In this paper, we propose a clique-based clustering to overcome the above limitations. First we define a metric to quantify the geometric overlap between two viewports on the sphere (Section II).

This work has been supported by Royal Society under grant IES\R1\180128 and by Adobe under Academic Donation scheme.

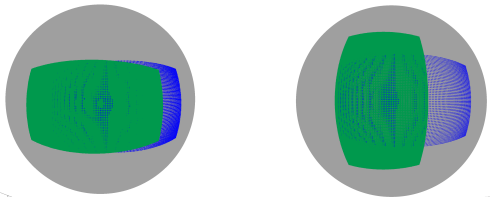
Then, we use this metric to build a graph whose nodes are the centers of the viewports associated to different users. Two nodes are connected only if the two corresponding viewports have a significant overlap (Section III). Finally, we propose a clustering method based on the Bron-Kerbosch (BK) algorithm [22] to identify clusters that are cliques, *i.e.*, sub-graphs of inter-connected nodes (Section III). Results demonstrate the consistency of the proposed clustering method in identifying clusters where the overlap between the portions of the spherical surface corresponding to different viewports is higher than in state-of-the-art clustering (Section IV). In summary, the main contribution of this paper is to propose a clustering algorithm that $i$) considers the spherical geometry of the data, $ii$) identifies clusters in which there is a consistent and significant geometric overlap between the portions of spherical surface corresponding to viewports attended by different users (by imposing that clusters are cliques), $iii$) can be applied to a single frame or to a series of frames (trajectories). This is a useful new tool to improve the accuracy of user's navigation prediction algorithms and user-dependent VR content delivery strategies, such as those proposed in [20, 21].

## 2. GEODESIC DISTANCE AS PROXY OF VIEWPORT OVERLAP

A key aspect of our clustering algorithm is to group users based on a metric that reliably reflects users similarities during the navigation. We argue that similarity in the navigation is captured by viewport overlap. In this section, we identify a metric that reliably reflects this overlap. More specifically, each user attends a portion of the spherical surface. This is the projection on the spherical surface of a plane tangent to the sphere (*i.e.*, *viewport*) in the point that identifies the user's viewing direction (*center of the viewport*)[1]. The overlap between the viewports attended by two users at an instant in time is a clear indicator of how similar users are with respect to their displayed viewports. For example, an overlap equal to the area of the viewport corresponds to two users attending exactly the same portion of visual content. The geometric overlap could be analytically computed, knowing the rotation associated to each user head's position (*i.e.*, roll, pitch, and yaw) and the horizontal and vertical fields of view that define the viewport. However, this is non trivial since it requires to evaluate closed-form expression of the viewport on the sphere. Here, we show that the *geodesic distance* between two viewport centres under specific settings acts as proxy of the viewport overlap. Thus, we propose the simple and straightforward solution of using this distance as a proxy for viewport overlap.

The geodesic distance is the length of the shortest arc connecting the viewport centers on the sphere. Such distance is an approximation of the actual area overlap as it does not account for the three degrees of freedom of the user's head rotation, which defines the exact viewport. As a result, viewports whose centers have the same geodesic distance could correspond to a different viewport overlap due to the intrinsic approximation error (example in Fig. 1). Nevertheless, the smaller the distance between viewport centers, the smaller the approximation error with geodesic distance. As an example, Fig. 2 shows the pairwise geodesic distance (in blue) and the pairwise area overlap (in red) between the viewport attended by one user and those of 58 other users, for a frame of a video sequence, extracted from the public dataset proposed in [13]. The correlation between the two metrics is evident: if the overlap is high, the geodesic distance between the two viewport centres is low. Particularly, a viewport area overlap that is larger than 75% of the viewport area corresponds to a geodesic distance smaller than $3\pi/4$. We are therefore interested in identifying a threshold value below which the geodesic distance is a robust proxy of the viewports overlap.

[1] Without loss of generalization, we consider a scenario in which the viewports of all users have the same horizontal and vertical field of view.



(a) 87% overlap (b) 58% overlap

**Fig. 1**. Viewports (in green and blue) with $\pi/10$ centre distance. (a) viewports are aligned with an overlap of 87%, (b) one viewport is rotated by $\pi/2$ resulting an overlap of 58%.

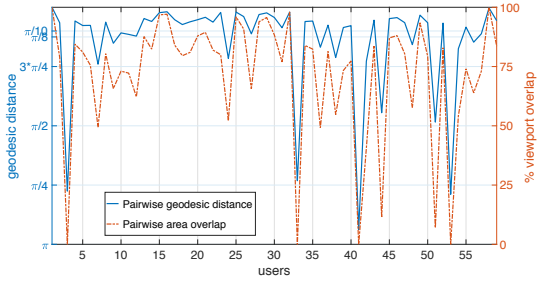



**Fig. 2**. Comparison between pairwise geodesic distance and viewport overlap in one frame of video Rollercoaster from [13].

To empirically define this threshold, we built the Receiver Operating Characteristic (ROC) curve as follows. We assume that two users are attending the same portion of content if their viewports overlap by at least $O_{th}$ of the total viewport area. We then define a threshold value for the geodesic distance $G_{th}$ such that users are attending the same content if their geodesic distance is below threshold. Anytime users are separeted by a geodesic distance lower than $G_{th}$ and the overlap of their viewport is less than $O_{th}$, we experience a false positive. Conversely, a true positive is experienced if users separated by a geodesic distance above the threshold but experience a viewport overlap equal or higher than $O_{th}$. Equipped with these definitions, we can compute the ROC by considering all the videos and user' navigation patterns included in the dataset described in [13]. Fig. 3 shows the curve obtained in our scenario with $O_{th} = 80\%$. On the x axis of the ROC curve there is the False Positive Rate (FPR), *i.e.*, the probability to have a wrong classification over the number of actual negative events. This rate should be as small as possible. On the contrary, the True Positive Rate (TPR) on the y axis represents the probability to correctly classify an event. The best value of geodesic distance is $\pi/10$ since it corresponds to a TPR value equal to 1, which in our application means a sure identification of viewports with an overlap of at least 80% based on the geodesic distance between their centers. Therefore, in the following we assume $G_{th} = \pi/10$ as a suitable threshold to reliably approximate the area overlap between two viewports by means of the geodesic distance between their centers.

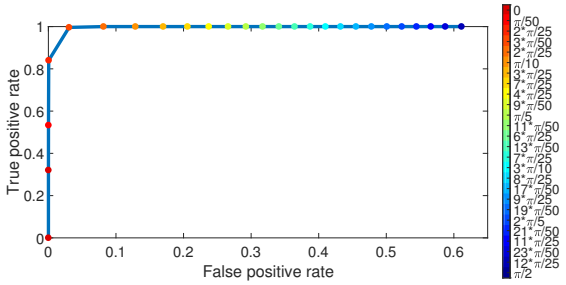



**Fig. 3**. ROC curve to evaluate optimal $G_{th}$ considering all video in database [13] and $O_{th} = 80\%$ .

## 3. CLIQUE-BASED CLUSTERING ALGORITHM

We now describe the proposed clustering algorithm, aimed at identifying clusters of users having a common viewport overlap. We model the evolution of users' viewports over a time-window $T$ as a set of graphs $\{\mathcal{G}_t\}_{t=1}^{T}$. Each unweighted and undirected graph $\mathcal{G}_t = \{\mathcal{V}, \mathcal{E}_t, \mathrm{W}_t\}$ represents the set of users[2] navigating over time, where $\mathcal{V}$ and $\mathcal{E}_t$ denote the node and edge sets of $\mathcal{G}_t$. Each node in $\mathcal{V}$ corresponds to a user interacting with the 360° content at instant $t$. Each edge in $\mathcal{E}_t$ connects neighbouring nodes, where two nodes are neighbours if the geodesic distance between the viewport centers associated to the users represented by the nodes is lower than $G_{th}$ , as defined in Section II. The binary matrix $\mathrm{W}_t$ is the adjacency matrix of $\mathcal{G}_t$, with $w_t(i,j) = 1$ if users are neighbors. More formally:

$$w_t(i,j) = \begin{cases} 1, & \text{if } g(i,j) \leq G_{th} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $g(i,j)$ is the geodesic distance between the viewport centres of users $i$ and $j$ and $G_{th}$ is thresholding value, introduced in Section II.

Looking at the graphs over time $\{\mathcal{G}_t\}_{t=1}^{T}$, we are interested in clustering users based on their trajectories within a time window of duration $T$. In other words, we are interested in identifying users that have similar behaviour over time. With this goal in mind, we derive an affinity matrix A that will be the input to our clustering algorithm Similarly to other clusters of trajectories [23]. Each element of A is defined as following:

$$a(i,j) = \mathcal{I}_\tau \left( \sum_{t=1}^{T} w_t(i,j) \right) \tag{2}$$

where the function $\mathcal{I}_\tau(\cdot)$ is defined as $\mathcal{I}_\tau(x) = 1$ if $x \geq \tau$ and 0 otherwise. The matrix A can be associated to a trajectory-based graph where two nodes $i$ and $j$ are neighbours only if the corresponding viewports have a significant overlap in $\tau$ instants over $T$, *i.e.*, $a(i,j) = 1$. The more threshold $\tau$ approaches $T$, the more stringent the similarity condition.

As clusters, we want to identify group of users that are all neighbours (*i.e.*, $a(i,j) = 1$ for all pairs of users $i$ and $j$ belonging to the cluster). In graph theory, a set of nodes that are all connected to each other is called a *clique*. A clique perfectly matches with our definition of meaningful cluster: set of users all having significant pairwise viewport overlap, thus attending a common portion of video. Therefore, we propose a *clique-based clustering*. In particular, we consider the *Bron-Kerbosch (BK) algorithm* [22] to find all *maximal cliques* present in our graph (*i.e.*, the most populated subgraphs forming cliques). While the BK algorithm identifies overlapping cliques (one user can belong to more than one clique), we are rather interested in identifying disjoint sets[3]. Hence, we build upon the BK algorithm and propose a clustering algorithm aimed at identifying non overlapping cliques, as depicted in Fig. 4. We initialize the clustering method by evaluating the affinity matrix from Eq. (2). Then, we perform the following steps (Algorithm 1):

1. Maximal cliques in the graph are detected by the BK algorithm.
2. Among the resulting cliques, only the most populated one (with the highest cardinality) is kept as a cluster.
3. A new affinity matrix is built, eliminating the entries corresponding to the elements of the cluster identified in Step 2.

These three steps are repeated until all nodes are assigned to clusters. It is worth mentioning that this iterative selection does not guarantee optimal clusters (*i.e.*, maximal joint overlap within the cluster). However, $i$) it imposes viewport overlap among users within a cluster, $ii$) it identifies highly populated clusters, which can be translated in reliable trajectories/behaviours shared among users.

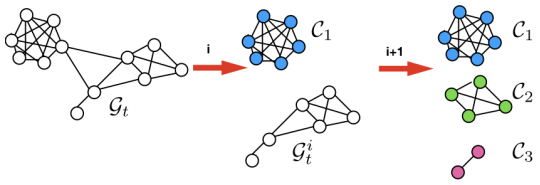



**Fig. 4**. Graphical example of the proposed clique clustering.

**Algorithm 1** Clique-Based Clustering

**Input:** $\{\mathcal{G}_t\}_{t=1}^{T}$, D
**Output:** $K, \boldsymbol{Q} = [Q_1, ..., Q_K]$
**Init:** $i = 1, \mathrm{A}^{(1)} = \mathcal{I}_D(\sum_t W_t), \boldsymbol{Q} = [\{\emptyset\}, \ldots, \{\emptyset\}]$
**repeat**
  $\boldsymbol{\mathcal{C}} = [\mathcal{C}_1, ..., \mathcal{C}_L] \leftarrow KB(\mathrm{A}^{(i)})$
  $l^\star = \arg\ \max_l |\mathcal{C}_l|$
  $Q_i = \mathcal{C}_{l^\star}$
  $\mathrm{A}^{(i+1)} = \mathrm{A}^{(i)}(\boldsymbol{\mathcal{C}} \setminus \mathcal{C}_{l^\star})$
  $i \leftarrow i + 1$
**until** $A^{(i)}$ *is not empty*;
$K = i - 1$

## 4. EXPERIMENTAL RESULTS

The proposed clustering algorithm is compared to state-of-the-art solutions, namely the *Louvain method* [24], the *K-means clustering* [25] and the clustering of VR trajectories proposed in [21] (labelled "SC"). We use the geodesic distance between viewport centers as distance metric in all algorithms. Moreover, in the *K-means clustering*, the number of clusters $K$ is imposed as the value achieved by

[2]Without loss of generality, we assume that the set of users does not change over time. This covers also cases in which users' devices are not synchronized in the acquisition time, as users' positions are usually interpolated to create a synchronized dataset.

[3]Clusters should be disjoint for most content-delivery applications. For example, if clusters are used for prediction, each user must belong only to one cluster.

| | | ROLLERCOASTER | | | | TIMELAPSE | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *Louvain method* | *Clique Clustering* | *K-Means 1* | *K-Means 2* | Louvain | *Clique Clustering* | *K-Means 1* | *K-Means 2* |
| Fr. ~30s | K | 10 | 15 | 10 | 15 | 13 | 24 | 13 | 24 |
| | Mean Overlap Cl.(% user >3) | 38.90 % (84.75 %) | **62.50 % (76.30 %)** | 53.95 % (93.20 %) | 48.10 % (94.90 %) | 46 % (89.70%) | **72.35% (56.90%)** | 45.90% (96.50 %) | 51.50% (50%) |
| | Main cl. overlap (% users) | 26.70% (44.10%) | **58.60% (30.50%)** | 48.30% (19% ) | 0% (20.70% ) | 32.90% (20.70%) | **69% (12.10% )** | 15% (19% ) | 23.50% (13.80%) |
| Fr. ~40s | K | 8 | 15 | 8 | 15 | 18 | 27 | 18 | 27 |
| | Mean Overlap Cl.(% users >3) | 35.60% (89.83%) | **65.75% (76.30%)** | 44.38% (100%) | 47.65% (84.75%) | 47.65 % (75.90%) | **72.95 % (77.60%)** | 60.27% (96.55%) | 65.90% (84.50%) |
| | Main cl. overlap (% users) | 24.20% (45.80%) | **58.33% (35.60%)** | 0% (30.50%) | 0% (15.25%) | 51.80% (20.70%) | **63.70% (17.24%)** | 47.50% (20.70%) | 33.60% (8.60%) |
| Fr. ~50s | K | 8 | 12 | 8 | 12 | 18 | 29 | 18 | 29 |
| | Mean Overlap Cl.(% users >3) | 48.20% (89.80%) | **65.70% (86.45%)** | 43.50% (98.30%) | 55.30% (96.60%) | 49.12 % (77.60%) | **71.40% (51.70%)** | 48.36 % (87.90%) | 55.90 % (55.17%) |
| | Main cl. overlap (% users) | 46.40%(30.50 %) | **59.90% (57.70%)** | 0% (22.40%) | 0% (15.25%) | 30.60 (22.40%)% | **70.80% (25.90%)** | 37% (24.15%) | 62.71% (17.24%) |

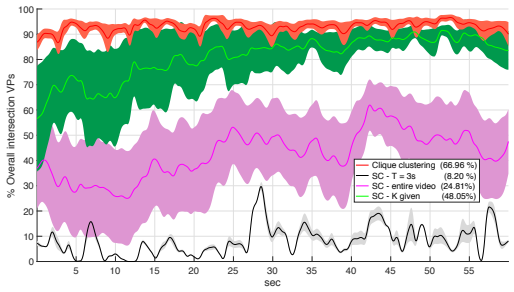
**Table 1**. Clustering analysis of users in three selected frames from Rollercoaster (first half) and Timelapse (second half). In brackets, the percentage of covered population.

the Louvain method (labelled "K-means 1"), as well as the $K$ value obtained from our proposed clustering (labelled "K-means 2"). The proposed implementations have been made publicly available [4]. We test these algorithms on two video sequences 1-minute long (Rollercoaster and Timelapse), which have been watched by 59 users whose navigation paths are publicly available [13]. Rollercoaster has one main RoI (i.e., the rail) while in Timelapse, there are many fast moving objects (e.g., buildings, people) along the equator line.

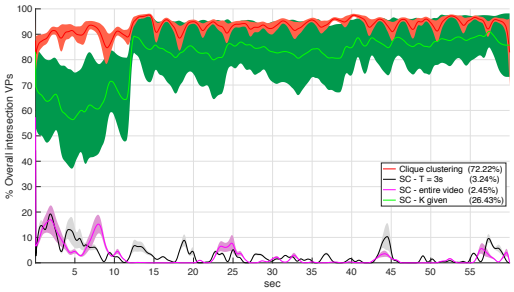
**Frame-based Clustering.** First, we consider frame-based clustering, in which users are identified by their viewport centers in one given frame. Table 1 reports results in terms of number of clusters ($K$), mean viewport overlap computed within each cluster composed by at least three users, and viewport overlap within the most populated cluster, that we refer to as the main cluster. The viewport overlap within a cluster is the joint overlap across all users' viewports in the cluster. The mean overlap is computed by averaging the viewport overlap of all clusters with at least three users identified at a given frame. In Table 1, we also provide the percentage of users covered by clusters. The proposed algorithm always ensures the highest viewport overlap (on average always over 50%) with respect to the other methods. This is due to the implicit constraint that is imposed by the clique-based detection of the clusters. This constraint leads to the identification of clusters that are populated and yet meaningful (i.e., with large viewport overlap among users). For example, in Rollercoaster at frame $40s$, our algorithm identifies a main cluster grouping $35\%$ of the population with a viewport overlap of $58.33\%$. This is much higher than the overlap of $24.20\%$ ($0\%$) in the main cluster identified by the Louvain (K-means) method. Beyond the accuracy, another important parameter is the percentage of the population that is covered by clusters with a significant number of users. These clusters are the most useful ones to allow predictions. For instance in Timelapse at frame $50s$, our method identifies a large number of clusters (29), which also includes single users clusters. Nevertheless, half of the population (51.70%) belongs to clusters with more than 3 users with high value of joint overlap (71.40%).

**Trajectory-based clustering.** Second, we test the proposed algorithm over a time-window of duration $T = 3s$ and similarity threshold $\tau = 1.8s$. In this case, we compare the proposed solution with SC algorithm [21]. The latter is applied in the following conditions: $i$) to trajectories spanning the entire video as in [21], $ii$) consecutive time windows of duration $T$ and $iii$) imposing the same $K$ obtained from our solution ("SC - K given"). Fig. 5 shows results in terms of overlap among viewports clustered together in both Rollercoster (a) and Timelapse (b). In more details, all users are clustered over consecutive time-windows of $T$ seconds each. Then, for each frame the viewport overlap among all users within one cluster is evaluated and averaged across clusters. The mean overlap (solid line) and the variance (shaded area) is finally depicted in Fig. 5. Moreover, the mean value of joint overlap in clusters with more than three users across the entire video is shown in the legend. Our solution outperforms SC in terms of mean overlap but also in terms of variance. The latter shows the stability of our clustering method ensuring for each cluster a consistent overlap over time. Finally, the performance gain is significant also in terms of overlap in the most populated clusters (value provided in the legend).

[4]https://github.com/LASP-UCL/spherical-clustering-in-VR-content.




(a) Rollercoaster video - T = 3 s.




(b) Timelapse video - T = 3 s.

**Fig. 5**. Mean and variance of the joint overlap across clusters over time. In the legend, the mean value of joint viewport overlap of clusters with more than three users performed across the entire video.

## 5. CONCLUSIONS

In this paper, we proposed a novel graph-based clustering strategy able to detect meaningful clusters, *i.e.*, group of users consuming the same portion of a virtual reality spherical content. First, we derived a geodesic distance threshold value to reflect the similarity among users and then we built a clique-based clustering based on this metric. Results carried out on real VR user navigation patterns show that the proposed method identifies clusters with higher joint overlap than other state-of-the-art clustering methods. Future works will focus on the application of our method in the framework of adaptive streaming of VR videos and for the prediction of user navigation patterns.

## 6. REFERENCES


[1] A. Serrano, V. Sitzmann, J. Ruiz-Borau, G. Wetzstein, D. Gutierrez, and B. Masia, "Movie editing and cognitive event segmentation in virtual reality video," vol. 36, 2017.

[2] S. Rossi and L. Toni, "Navigation-Aware Adaptive Streaming Strategies for Omnidirectional Video," in *IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, 2017.

[3] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *IEEE International Conference on Communications (ICC)*, 2017.

[4] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "An HTTP/2-Based Adaptive Streaming Framework for 360 Virtual Reality Videos," in *Proceedings of the 2017 ACM on Multimedia Conference*, 2017.

[5] C. Fan, J. Lee, C. Lo, W.and Huang, K. Chen, and Cheng-H. H., "Fixation Prediction for 360 Video Streaming in Head-Mounted Virtual Reality," in *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2017.

[6] M. Yu, H. Lakshman, and B. Girod, "A Framework to Evaluate Omnidirectional Video Coding Schemes," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2015.

[7] M. Broeck, F. Kawsar, and J. F. Schöning, "It's all around you: Exploring 360-degree video viewing experiences on mobile devices," in *Proceedings of the 2017 ACM on Multimedia Conference (MM)*, 2017.

[8] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in VR: How Do People Explore Virtual Environments?," vol. 24, no. 4, April 2018.

[9] K. Srivastava, RC Das, and S Chaudhury, "Virtual reality applications in mental health: Challenges and perspectives," *Industrial psychiatry journal*, vol. 23, 2014.

[10] E. Upenik and T. Ebrahimi, "A simple method to obtain visual attention data in head mounted virtual reality," in *IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2017.

[11] B. Hu, I. Johnson-Bey, M. Sharma, and E. Niebur, "Head movements during visual exploration of natural images in virtual reality," in *IEEE 51st Annual Conference on Information Sciences and Systems (CISS)*, 2017.

[12] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A Dataset for Exploring User Behaviors in VR Spherical Video Streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys)*, 2017.

[13] X. Corbillon, F. De Simone, and G. Simon, "360-degree video head movement dataset," in *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys)*, 2017.

[14] C. Lo, W.and Fan, J. Lee, C. Huang, K. Chen, and C. Hsu, "360 Video Viewing Dataset in Head-Mounted Virtual Reality," in *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys)*, 2017.

[15] S. Fremerey, A. Singla, K. Meseberg, and A. Raake, "AVtrack360: an open dataset and software recording people's head rotations watching 360 videos on an HMD," in *Proceedings of the 9th ACM Multimedia Systems Conference (MMSys)*, 2018.

[16] A. Singla, S. Fremerey, W. Robitza, P. Lebreton, and A. Raake, "Comparison of Subjective Quality Evaluation for HEVC Encoded Omnidirectional Videos at Different Bit-rates for UHD and FHD Resolution," in *Proceedings of the ACM Conference on Multimedia (MM) Thematic Workshops*, 2017.

[17] A. Duchowski and G. Marmitt, *Modeling Visual Attention in VR: Measuring the Accuracy of Predicted Scanpaths*, Ph.D. thesis, 2002.

[18] E. J. David, J. Gutiérrez, A. Coutrot, M. Da Silva, and P. Le Callet, "A Dataset of Head and Eye Movements for 360° Videos," in *Proceedings of the 9th ACM Multimedia Systems Conference (MMSys)*, 2018.

[19] Y. Rai, P. Le Callet, and P. Guillotel, "Which saliency weighting for omni directional image quality assessment?," in *International Conference on Quality of Multimedia Experience (QoMEX)*, May 2017.

[20] L. Xie, X. Zhang, and Z. Guo, "CLS: A Cross-user Learning based System for Improving QoE in 360-degree Video Adaptive Streaming," in *ACM Multimedia Conference on Multimedia Conference (MM)*, 2018.

[21] S. Petrangeli, G. Simon, and V/ Swaminathan, "Trajectory-Based Viewport Prediction for 360-Degree Virtual Reality Videos," in *IEEE conference on Artificial Intelligence and Virtual Reality (AIVR)*, 2018.

[22] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, 1973.

[23] S. Atev, G. Miller, and N. P. Papanikolopoulos, "Clustering of vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, 2010.

[24] V. D. Blondel, J. Guillaume, and E. Lambiotte, R.and Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, , no. 10, 2008.

[25] J. Hartigan and M. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society*, vol. 28, 1979.