README.md

Viewport-aware Dynamic 360 Video Segment Categorization

This repository holds the code to reproduce the analysis and the proposed algorithms of the accepted paper #106 Viewport-aware Dynamic 360 Video Segment Categorization for NOSSDAV 2021.

Requirements

The following packages are required

- python 3.8.5
- numpy 1.19.2
- matplotlib 3.3.2
- pandas 1.2.1
- scikit-learn 0.23.2
- scikit-image 0.17.2
- scipy 1.6.0
- seaborn 0.11.1
- tqdm 4.56.0
- Shapely 1.7.1
- statsmodels 0.12.1

Datasets

The analysis is done for an aggregaed dataset comprising of six different public datasets of head-movement logs in viewing 360 videos. [1] , [2], [3], [4], [5], [6].

Please refer to our repository https://github.com/360VidStr/A-large-dataset-of-360-video-user-behaviour/ for more details on the dataset

The total number of videos in the aggregated dataset is 88. The preprocessed dataset is in the folder `./aggregated_dataset/` . Logs for the nth video is in `./aggregated_dataset/n.txt`

1. all datasets are resampled at 10Hz. t=[0,0.1,0.2...]
2. Video numbers were taken to name the files. The information about the videos are given in `./aggregated_dataset/VideoMetadata.xlsx`
3. Every file has 2n+1 lines where n= number of users who watched the video. First line contains the time points of sampling (in seconds).
4. 2i-1 th line has pitch angles for the ith user, 2i th line has the yaw angles for the ith user

Note

In this version, the defaul path has been set to `'/home/nossdavsub106/nossdav_artifacts/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/'` , if you want to change the folder placement, open `./setPath.py` and replace the `folder_path` by the new path to the folder.

This path will be set as the default `--path` argument in the following scripts.

Feature extraction

To implement the proposed feature extraction in Section 4.1, run the script `./feature_extraction.py` with the following arguments

1. `--path ~/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/` (Pass the complete path to the directory of the cloned repository)
2. `--video video_number` (specify a number 0-87 to extract features for a single video. If not set, the script extracts features for all 88 videos in the dataset)

Example command

```
python3 feature_extraction.py --path /home/nossdavsub106/nossdav_artifacts/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/ --video 10
```

The extracted features will be saved in the directory `./extracted_features/` (Results have been already generated and saved in the corresponding folders. Further verification can be done by running the script and overwriting the results)

Eg: The path to the extracted features of viewport chunks from 10-12 seconds of the 5th video is at `./extracted_features/10-12s/5.txt` . Where the nth line corresponds to the nth user. 17 space seperated values in each line corresponds to the following features in order

1. Video number (0-87)
2. User number
3. Yaw position - 25th percentile
4. Yaw position - mean
5. Yaw position - 75th percentile
6. Pitch position - 25th percentile
7. Pitch position - mean
8. Pitch position - 75th percentile
9. Percentage of sphere explored
10. Maximum anglular displacement from starting point - pitch
11. Maximum anglular displacement from starting point - yaw
12. Speed in yaw - 25th percentile
13. Speed in yaw - mean
14. Speed in yaw - 75th percentile
15. Speed in pitch - 25th percentile
16. Speed in pitch - mean
17. Speed in pitch - 75th percentile

A file `./extracted_features/allFeatures.txt` will be saved with features of all viewport chunks. A line in this file corresponds to the features extracted from a 2s chunk of viewport logs. 18 space seperated values in each lines are same as above with the starting second of the viewport chunk inserted between 2. User number and 3. Yaw position - 25th percentile.

Viewport clustering for the aggregated dataset

To implement the proposed analysis for the optimum nnumber of clusters for the aggregated datset, run the script `./optimum_VP_cluster_number_analysis.py` with the following arguments

1. `--path ~/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/` (Pass the complete path to the directory of the cloned repository)

The resulting analysis (showed in Figure 5.a of the paper) will be saved to `./optimum_VP_cluster_number_analysis_results` . According to the explanation given in the paper, the optimum number of clusters is selected as 10.

Example command

```
python3 optimum_VP_cluster_number_analysis.py --path /home/nossdavsub106/nossdav_artifacts/Viewport-Aware-Dynamic-360-Video-
Segment-Categorization/
```

To implement the proposed viewport clustering for the aggregated dataset, run the script `./viewport_clustering_for_dataset.py` with the following arguments

1. `--path ~/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/` (Pass the complete path to the directory of the cloned repository)
2. `--nclusters n` (number of viewport clusters. default is 10)
3. `--saveClusters` (set this to save the elemets of resulting clusters to text files)
4. `--savePlots` (set this to save the plots in figure 7)
5. `--eval` (set this to run the evaluation explained in Section 4.2 and save the figure 6)

Example command

```
python3 viewport_clustering_for_dataset.py --path /home/nossdavsub106/nossdav_artifacts/Viewport-Aware-Dynamic-360-Video-
Segment-Categorization/  --nclusters 10 --saveClusters --savePlots --eval
```

The clustering and evaluation results will be saved to a folder named after the `--nclusters` passed to the script. Eg: `./viewport_clustering_for_dataset_results/nclusters_10/` .
The elements of the resulting clusters will be saved to text files named after the cluster number Eg: `./viewport_clustering_for_dataset_results/nclusters_10/cluster_0.txt` . Each line will include a viewport chunk characterized by `videoNumber UserNumber startingTime(s)`

Comparing Viewport clustering algorithms

The precomputed results of viewport clustering (for the set of viewport chunks of the users watching the same video at the same time) using Spherical-clustering [7], DBScore-based clustering [9], and Trajectory-based clustering [8] are saved in the folders `./sphericalClusteringResults/` , `./DBClusteringResults/` , `./trajectoryClusteringResults/` respectively.

To implement the comparison of the proposed algorithm with the results of Spherical-clustering [7], DBScore-based clustering [9], and Trajectory-based clustering [8] and reproduce the results of Figure 3, run the script `./comparing_vp_clustering_algorithm.py` with the following arguments

1. `--path ~/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/` (Pass the complete path to the directory of the cloned repository).

Example command

```
python3 comparing_vp_clustering_algorithm.py --path /home/nossdavsub106/nossdav_artifacts/Viewport-Aware-Dynamic-360-Video-
Segment-Categorization/
```

The results will be saved to `./comparing_vp_clustering_algorithm_results/`

Dynamic Video Segement Categorization for the aggregated dataset

To implement the proposed analysis for the optimum nnumber of video categories for the aggregated datset, run the script `./optimum_video_categories_number_analysis.py` with the following arguments

1. `--path ~/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/` (Pass the complete path to the directory of the cloned repository)
2. `---viewportClusters n` (number of viewport clusters. default is 10)

Example command

```
python3 optimum_video_categories_number_analysis.py --path /home/nossdavsub106/nossdav_artifacts/Viewport-Aware-Dynamic-360-
Video-Segment-Categorization/ --viewportClusters 10
```

The resulting analysis (Figure 10.a) will be saved to `./optimum_video_categories_number_analysis_results` . According to the explanation given in the paper, the optimum number of video categories corresponding to 10 viewport clusters is selected as 6.

To implement the proposed video categorization for the aggregated dataset, run the script `./video_categorization_all.py` with the following arguments

1. `--path ~/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/` (Pass the complete path to the directory of the cloned repository)
2. `--viewportClusters n` (number of viewport clusters. default is 10)
3. `--videoCats v` (number of video categories. default is 6)
4. `--saveClusters` (set this to save the elemets of resulting clusters to text files)
5. `--savePlots` (set this to save the plots in figure 9)
6. `--eval` (set this to run the evaluation explained in Section 4.2 and save the figure 10.b)

Example command

```
python3 video_categorization_all.py --path /home/nossdavsub106/nossdav_artifacts/Viewport-Aware-Dynamic-360-Video-Segment-
Categorization/ --viewportClusters 10 --videoCats 6 --saveClusters --savePlots --eval
```

The results of the categorization and evaluation will be saved to a folder named after the arguments passed for `---viewportClusters` and `---videoCats` . Eg: `./video_categorization_all_results/VPclusters_10_videoCats_6/` . The plot corresponding to Figure 9 and evaluation results corresponding to 10.b will be saved in the results folder.
The text files corresponding to the video category number contains in each line, `VideoNumber startTimeOfVideoSegment(s)` .

Dynamic Video Segement Categorization vs Genre-based video categorization

To reproduce the comparison analysis given in Figure 11.b, run the script `./genre_vs_dynamic_video_categorization.py` with the following arguments

1. `--path ~/Viewport-Aware-Dynamic-360-Video-Segment-Categorization/` (Pass the complete path to the directory of the cloned repository)
2. `--viewportClusters n` (number of viewport clusters. default is 10)
3. `--videoCats v` (number of video categories. default is 6)

Example command

```
python3 genre_vs_dynamic_video_categorization.py --path /home/nossdavsub106/nossdav_artifacts/Viewport-Aware-Dynamic-360-
Video-Segment-Categorization/ --viewportClusters 10 --videoCats 6
```

The 88 videos of the dataset are categorized to 10 genres

0. Animals
1. cartoon
2. concert
3. documentary
4. driving
5. rollercoaster
6. scenery
7. shark
8. sports
9. video game

Results will be saved to `./genre_vs_dynamic_video_categorization_results/`

References

[1] Xavier Corbillon, Francesca De Simone, and Gwendal Simon. 2017. 360-Degree Video Head Movement Dataset. In Proceedings of the 8th ACM on Multimedia Systems Conference. ACM, Taipei Taiwan, 199–204. https://doi.org/10.1145/3083187.3083215

[2] Wen-ChihLo, Ching-LingFan, Jean Lee, Chun-Ying Huang, Kuan-Ta Chen,and Cheng-Hsin Hsu.2017. 360° Video Viewing Dataset in Head-Mounted Virtual Reality.In Proceedings of the 8th ACM on Multimedia Systems Conference.ACM, Taipei Taiwan, 211–216. https://doi.org/10.1145/3083187.3083219

[3] Yanan Bao, Huasen Wu, Tianxiao Zhang, Albara Ah Ramli, and Xin Liu.2016. Shooting a moving target: Motion-prediction-based transmission for 360-degree videos. In 2016 IEEE International Conference on Big Data (Big Data). IEEE,Washington DC, USA ,1161–1170. https://doi.org/10.1109/BigData.2016.7840720

[4] Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang. 2017. A Dataset for Exploring User Behaviors in VR Spherical Video Streaming. In Proceedings of the 8th ACM on Multimedia Systems Conference. ACM, Taipei Taiwan, 193–198. https://doi.org/10.1145/3083187.3083210

[5] Yu Guan, Chengyuan Zheng, Xinggong Zhang, Zongming Guo, and Junchen Jiang.2019. Pano: optimizing 360° video streaming with a better understanding of quality perception. In Proceedings of the ACM Special Interest Group on Data Communication. ACM, Beijing China,394–407. https://doi.org/10.1145/3341302.3342063

[6] Afshin Taghavi Nasrabadi, Aliehsan Samiei, Anahita Mahzari, Ryan P. McMahan, Ravi Prakash, Mylène C. Q. Farias, and Marcelo M. Carvalho. 2019. A taxonomy and dataset for 360° videos. In Proceedings of the 10th ACM Multimedia Systems Conference. ACM, Amherst Massachusetts, 273–278. https://doi.org/10.1145/3304109.3325812

[7] S. Rossi, F. DeSimone, P. Frossard, and L. Toni. 2019.Spherical Clustering of Users Navigating 360° Content.In ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).4020–4024. https://doi.org/10.1109/ICASSP.2019.8683854 ISSN:2379-190X.

[8] Stefano Petrangeli, Gwendal Simon, and Viswanathan Swaminathan. 2018. Trajectory-Based Viewport Prediction for 360-Degree Virtual Reality Videos. In2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR).IEEE,Taichung,Taiwan,157–160. https://doi.org/10.1109/AIVR.2018. 00033

[9] Lan Xie, Xinggong Zhang, and Zongming Guo. 2018. CLS:A Cross-user Learning based System for Improving QoE in 360-degree Video Adaptive Streaming. In 2018 ACM Multimedia Conference on Multimedia Conference-MM'18.ACMPress, Seoul,Republic of Korea, 564–572. https://doi.org/10.1145/3240508.3240556