Task Description: Operation Periscope

Celtra’s Programming Challenge 2017

Task: Operation Periscope

Task Description

The viewer of a 360° video is positioned in the center and the video is played all around him/her. While watching the video, the viewer is able to change his/her direction by turning the mobile phone, swiping on the video or, in case of using VR goggles, turning his/her head. This makes it possible for the viewer to focus on the interesting parts of the video.

Due to high quality of videos and limited bandwidth (especially when using mobile networks) it is not feasible to download the whole video in full quality. The task is to optimize the amount of data downloaded to the viewer's device. The quality of video shown to the user should be **as high as possible** while keeping in mind his viewing angle and **bandwidth limitation**.

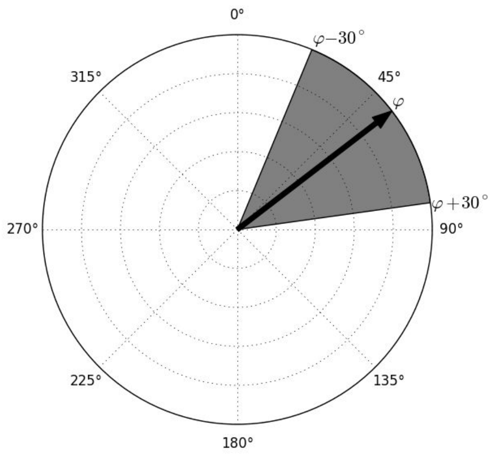

Figure 1 - Arrow represents the viewer's direction and the gray area represents the viewing angle.

The original video clip covers all 360° degrees (full circle) around the user. **The first part of this task** is to divide the original video clip into streams. Each stream $s_i = (\alpha_i, \beta_i, q_i)$ covers the angle from α to β (inclusive) in certain quality $q \in [1, 100]$. **The second part** is to implement an algorithm, which selects a set of video streams that will be preloaded at specific times in order to be able to play the video to the viewer. For this task, the algorithm should be able to use the trajectories of other viewers and the trajectory of the current user to predict which video streams will be present in the viewing angle. The video displayed on the viewer's device can be composed from several streams. In case several streams overlap, the highest-quality stream is selected for the overlapping area. The goal is to maximize the quality of video played on the viewer's device during the full duration of the video clip.

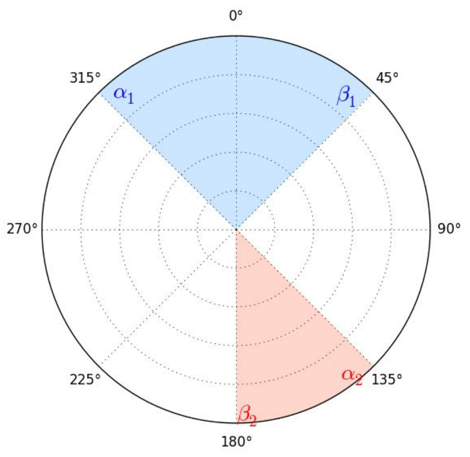

Figure 2 - The blue and red areas represent the ranges of two sample video streams.

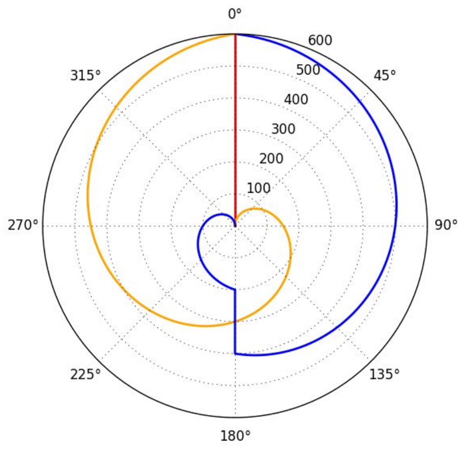

Figure 3 - The figure shows 3 simple trajectories. The time runs from the center of the graph towards the outer limit.
Red: The viewer never changed the view direction.
Yellow: The viewer steadily turns anticlockwise.
Blue: The viewer turns for 180° clockwise, then after a while continues turning clockwise.

Limitations:

- The number of video streams must not exceed 100.
- The viewing angle is 60°.
- The amount of data downloaded to the viewer’s device must not exceed the bandwidth limitation $B = 10000$ at any moment. We estimate the amount of data needed to download a single video stream as $(\beta_i - \alpha_i) * q_i$
- The algorithm must ensure that at least a minimum-quality stream $q \geq 1$ is available for the whole 360° at all times.

The **learning dataset** is given in CSV format. It is composed of a number of trajectories that were recorded while user viewed the 360° video of length T.

A sample of learning data (i-th trajectory):

Trajectory ID	**Time**	**Viewer's direction**
$traj_i$	0	$\varphi_{i,0}$
$traj_i$	1	$\varphi_{i,1}$
...	...	...
$traj_i$	T	$\varphi_{i,T}$

Test dataset includes a number of partial trajectories - up to time $t_{j,m}$. The goal of this task is to preload the right video streams at the right moment, i.e. predict the set of video streams at times from $t_{j,m+1}$ to $t_{j,m+n}$ that make it possible to play the video in as high as possible quality.

Sample test data (j-th partial trajectory):

Partial trajectory ID	**Time**	**Viewer's direction**
$traj_j$	$t_{j,0}$	$\varphi_{j,0}$
$traj_j$	$t_{j,1}$	$\varphi_{j,1}$
...	...	...
$traj_j$	$t_{j,m}$	$\varphi_{j,m}$
$traj_j$	$t_{j,m+1}$	NA
$traj_j$	$t_{j,m+2}$	NA
...	...	...
$traj_j$	$t_{j,m+n}$	NA

Submission

Solution:

- CSV file containing the video stream definitions
- CSV file containing the predictions for all partial test trajectories

Program code:

- Use whichever programming language best suits you

Report:

- 2-5 A4 pages
- Justify your approach
- Describe your solutions and results

Evaluation criteria

Predictions for a single test trajectory are evaluated: $score = \sum_{t=m+1}^{m+n} \sum_{\alpha=\varphi(t)-30}^{\varphi(t)+30} max_{i \in S(t)}(q_i(\alpha))$

$\varphi(\tau)$ … the viewer's direction at time τ
$S(\tau)$ … the set of video streams that are preloaded at time τ
$q_i(\alpha)$ … the quality of video stream S_i at angle α

If any of the limitations are violated the $score = 0$:

- $|S| \leq 100$
- $\forall t: \sum_{i \in S(t)} (\beta_i - \alpha_i) * q_i \leq$ 10000
- $\forall t,\ \forall \alpha : max_{i \in S(t)}(q_i(\alpha)) \geq 1$

The final submission evaluation is calculated as the sum of evaluations for all test trajectories.

Skills needed

- Data pre-processing
- Data mining / machine learning
- Optimisation methods