Short description

Takes multiple video files (.avi) and outputs one video file containing all streams in each order. Can, for instance, be used to decrease data size if multiple cameras are installed in one enclosure. Up to six video streams can be merged, such that the ordering is as follows:

<table>
<tr><th>2 streams</th><th>3 or 4 streams</th><th>5 or 6 streams</th></tr>
<tr>
<td>
<table>
<tr><td>Stream 1</td><td>Stream 2</td></tr>
</table>
</td>
<td>
<table>
<tr><td>Stream 1</td><td>Stream 2</td></tr>
<tr><td>Stream 3</td><td>Stream 4 / black region</td></tr>
</table>
</td>
<td>
<table>
<tr><td>Stream 1</td><td>Stream 2</td><td>Stream 3</td></tr>
<tr><td>Stream 4</td><td>Stream 5</td><td>Stream 6 / black region</td></tr>
</table>
</td>
</tr>
</table>

Requirements

- packages: opencv, numpy

Step 1 - open spyder:

- shell / terminal:
  - conda activate bovids
  - spyder

Step 2 – adjusting parameters:

- OUTPUT_DESTINATION= destination path (ends with .avi), all folders of this path need to exist. [string]
- OUTPUT_FPS: fps of the output video file, to use BOVIDS, this needs to be set to 1. [integer]
- VIDEOLISTE: Dictionary such that the keys are full paths to video files (.avi). The corresponding values are integers indicating the position of the stream in the output. [dictionary]

- E.g.: VIDEOLISTE = {
  'Z:/examples/2019-11-30_Wildebeest_FancyZoo_1.avi' : 2,
  'Z:/examples/2019-11-30_Wildebeest_FancyZoo_3.avi': 1
  }

Step 3 – run the script and enter

- ***merge_videos()***