OF-based background blurring

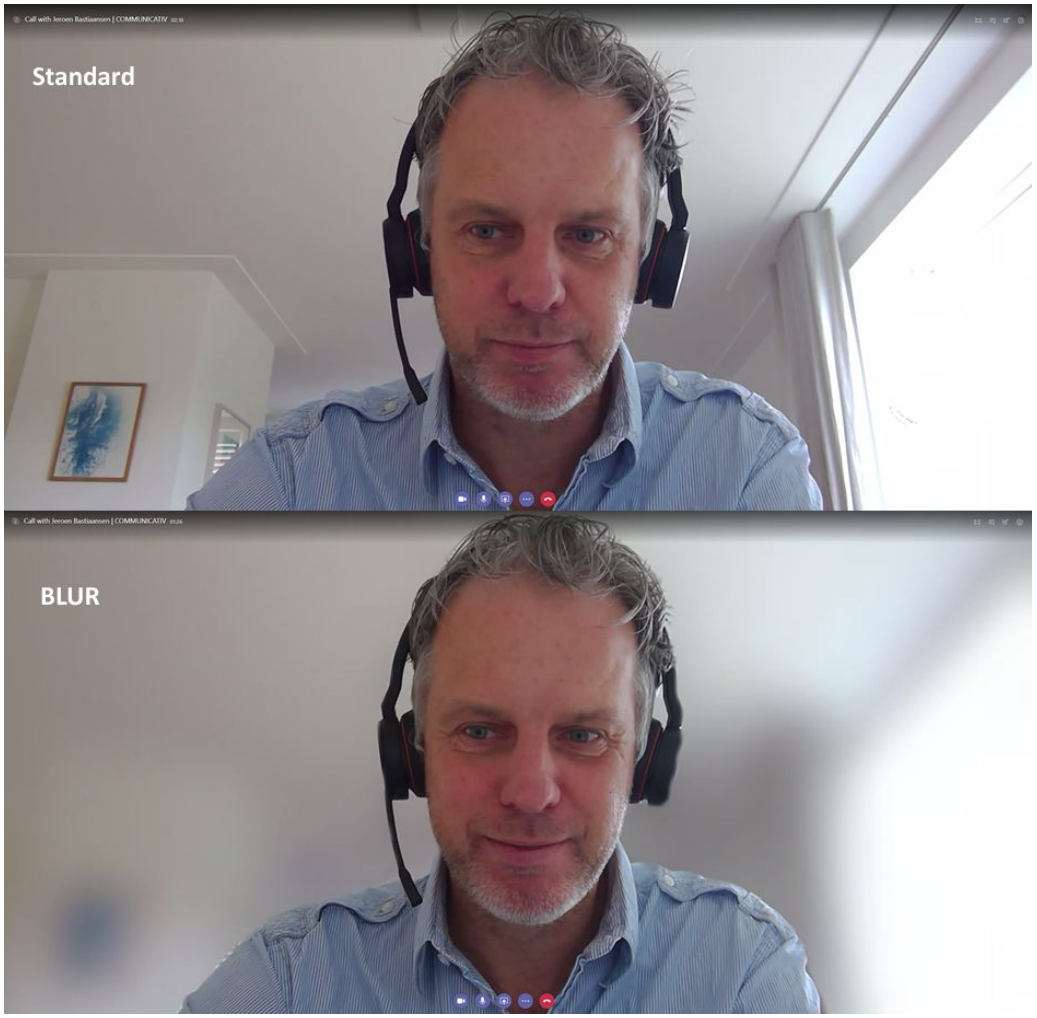
Objectives:

- To learn how to compute optical flow between a pair of frames.
- To learn how to use binary masks to manipulate the image content.
- To learn how to use the optical flow as a feature for content segmentation.
- To learn how to use the "running average" estimator to model a system memory.

For validating your code, use the videos provided in Moodle and others that you could record or find on the Internet.

1. Minimum (Up to 7/10 points)

Popular programs for video-calls, as Zoom or Skype, allow you to blur your background during the call. We propose in this practical assignment to create a simple program to modify the background pixels of an image, assuming that the original background is mostly static and the foreground person is moving.



As we assume that the background of the scene is static, we will use as a feature the magnitude of the optical flow vectors estimated between pairs of frames. The basic approach is to normalize in [0,1] the magnitude matrix obtained at each time stamp, and to threshold those values.

In summary, given an input video, the algorithm has to compute, for each pair of consecutive frames, the optical flow at each position of the frame. Then, based on a predefined criterion, a binary mask has to be generated, where the background pixels will be marked as 0. Finally, those pixels marked as 0 will be blurred, keeping the foreground pixels unchanged.

An incomplete source code file will be provided to the students, so they can start coding on that file. Do not forget to complete the TODO parts, in addition to the requested functions.

In order to pass this compulsory part, you have to implement the following functions:

- *fsiv_compute_dense_optical_flow*: given two frames, it computes their corresponding optical flow;
- *fsiv_compute_optical_flow_magnitude*: it computes the magnitude of the optical flow vectors at each position;
- *fsiv_compute_of_foreground_mask*: it computes a binary mask that can be used later to blur the unselected pixels. The basic way of generating this mask is by just normalizing the optical flow magnitude in the range [0,1];
- *fsiv_blur_background*: it applies blurring to the unselected pixels.

Useful OpenCV functions: *split, blur, getStructuringElement.*

Evaluation of the compulsory part.
IMPORTANT: in addition to the code, a file named “README.txt” has to be included describing how to compile the code and presenting at least three command-line examples of program calls. Screenshots of the outputs will be also taken into account.

Code is well-written, structured and compiled. OpenCV is correctly used.	+1
The code contains the basic functionality obtaining and applying a mask based on the normalized magnitude of the optical flow. It works both on camera streams and video files.	+6

(* Late delivery of this assignment will imply a penalty in the score of it. The score of each part has to be understood as “up to”).

2. Optional (Up to 3/10 points)

a) (+1) Add options (*i*) to binarize the mask by thresholding the OF magnitude and (*ii*) to apply a structuring element to dilate the mask.
b) (+1) Use memory (i.e. running average with an alpha factor) between frame updates. Basically, the new mask is computed as the weighted mean between the previous mask and the current optical flow magnitude:
   mask = alfa * old_mask + (1.0 - alfa) * new_mask
   The contribution of the old mask is defined by a float value in the range [0,1].
c) (+1) Add trackbars to change on-the-fly the magnitude threshold for binarizing the mask and other parameters such as the size of the structuring elements or the radius of the blurring filter.