Assignment 6

Welcome to the sixth assignment of the lecture *2D Vision and Deep Learning*. **Please read all instructions carefully!** This assignment covers motion estimation. Submission is due on Monday, November 14th, 2020 at 2pm. Please note that late assignments will receive zero (0) marks, so you are strongly encouraged to start the assignment early. If you have any questions please contact Tim Heußler (`theussle@students.uni-mainz.de`).

Exercise 1 (4 points)**.** Write a simple python OpenCV program that calculates and visualizes *difference images*. Thereby your program should determine the following two different types of *difference images*:

1. (2 points) The difference of two consecutive images $I_{t-\Delta t}$ and I_t (with threshold s):
$$I_t^{diff}(x,y) = \begin{cases} 1 & \text{if } |I_t(x,y) - I_{t-\Delta t}(x,y)| > s \\ 0 & \text{else} \end{cases}$$

2. (2 points) The difference of three consecutive images:
$$I_t^{and}(x,y) = \begin{cases} 1 & \text{if } I_t^{diff}(x,y) = 1 \wedge I_{t+\Delta t}^{diff}(x,y) = 1 \\ 0 & \text{else} \end{cases}$$

Your program should be able to read in a video stream either from the file system or a webcam. What is the advantage of using I_t^{and} compared to I_t^{diff} in order to detect and determine motion in an image sequence?

Exercise 2 (6 points)**.** Write a program that uses the mean optical flow of an image sequence to interactively control an application. Realize the following functionality:

1. (2 point) Downsample each image of the image sequence (or at least two consecutive images) using an image pyramid.

2. (2 point) Compute the *mean optical flow* between two consecutive images in the image sequence at a specific pyramid level (image resolution n), by solving
$$\begin{pmatrix} I_x(x_1,y_1,t) & I_y(x_1,y_1,t) \\ \vdots & \vdots \\ I_x(x_n,y_n,t) & I_y(x_n,y_n,t) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} I_t(x_1,y_1,t) \\ \vdots \\ I_t(x_n,y_n,t) \end{pmatrix}$$
The level to be selected should be a parameter of your program.

3. (2 points) Use the calculated optical flow to distinguish between the following six different hand motions in a video stream coming from a webcam or the file system: *Static (no hand motion)*, *Motion to the Left*, *Motion to the Right*, *Upward Motion*, *Downward Motion*.

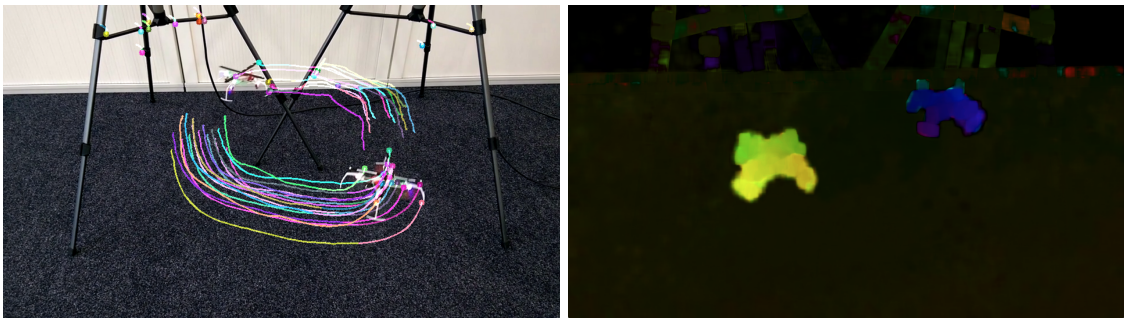
Exercise 3 (6 points)**.** Write a python OpenCV program to compare differend approaches to calculate the optical flow in a sequence of images. Basically, there are two different categories of algorithms to determine optical flow. One type like the method from *Lucas & Kanade*[1] computes optical flow for a sparse set of image features while the other type like the method from *Farnebäck*[2] or *Kroeger et. al.*[3] computes the optical flow for all pixels of an image sequence.

Realize different variants of both types of algorithms using

1. (2 points) the OpenCV function `calcOpticalFlowPyrLK` to determine the optical flow of a sparse set of points (use the function `goodFeaturesToTrack` to determine these points) (see figure 1 (a)),

2. (2 points) the OpenCV function `calcOpticalFlowFarneback` to calculate a dense field of optical flow (see figure 1 (b), where the angle and magnitude of the flow is color coded using the HSV color space).

3. (2 points) the OpenCV function `DISOpticalFlow` to calculate a denxe field of optical flow. Discuss the differences between the algorithm that is realized in `calcOpticalFlowFarneback` and that which is realized in `DISOpticalFlow`.



(a) Sparse optical flow (b) Dense optical flow

Figure 1: Sparse and dense optical flow on a video showing two flying quadcopter.

You can find the footage of the flying quadcopters used in figure 1 at `lms.uni-mainz.de/`.

[1]B. D. Lucas, T. Kanade, *An iterative image registration technique with an application to stereo vision.* Proceedings of Imaging understanding workshop, 1981, pp 121–130

[2]G. Farnebäck, *Two-frame Motion Estimation Based on Polynomial Expansion*, Proceedings of the 13th Scandinavian Conference on Image Analysis, 2003, pp 363–370

[3]T. Kroeger, R. Timofte, D. Dai, and L. V. Gool. Fast optical flow using dense inverse search. In Proceedings of the European Conference on Computer Vision (ECCV), 2016.