# CS 423/523 Introduction to Computer Vision - Assignment 3

**Due: 11th of January 23:55**

## Action Recognition with Optical Flow

## Description:

In this assignment, you will be classifying the action in a given video. In [this](http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html) link (http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html) you can download the dataset you will be working on. **Note that you need to download all of the classes in the Classification Database.** On this dataset, you will be applying HOOF [1] (Histogram of Oriented Optical Flow), PCA (Principal Component Analysis) and k-Nearest Neighbors (k-NN) for classification.

## Details:

First, you need to divide your dataset into training and test sets. Leave one test sample per each class and use others for training. Steps for finishing this assignment are as follows;

- create temporal mean HOOF for each training sample,
- apply PCA to HOOF and reserve 90% of the total variance,
- create temporal mean HOOF and apply PCA transform on training samples,
- classify via k-NN. Your goal is to achieve the best result by choosing **k** and **number bins** for HOOF.

In the paper, HOOF is defined as follows. First, optical flow is computed at every frame of the video. Each flow vector is binned according to its primary angle from the horizontal axis and weighted according to its magnitude. Thus, all optical flow vectors, $v = [x, y]^T$ with direction, $\theta = tan^{-1}\left(\frac{y}{x}\right)$ in the range,

$$-\frac{\pi}{2} + \pi\frac{b-1}{B} \leq \theta < -\frac{\pi}{2} + \pi\frac{b}{B}$$

will contribute by $\sqrt{x^2 + y^2}$ to the sum in bin $b$, $1 \leq b \leq B$ out of a total of $B$ bins. Finally, HOOF is normalized to sum up to 1. You can find an example of bins for $B = 4$,

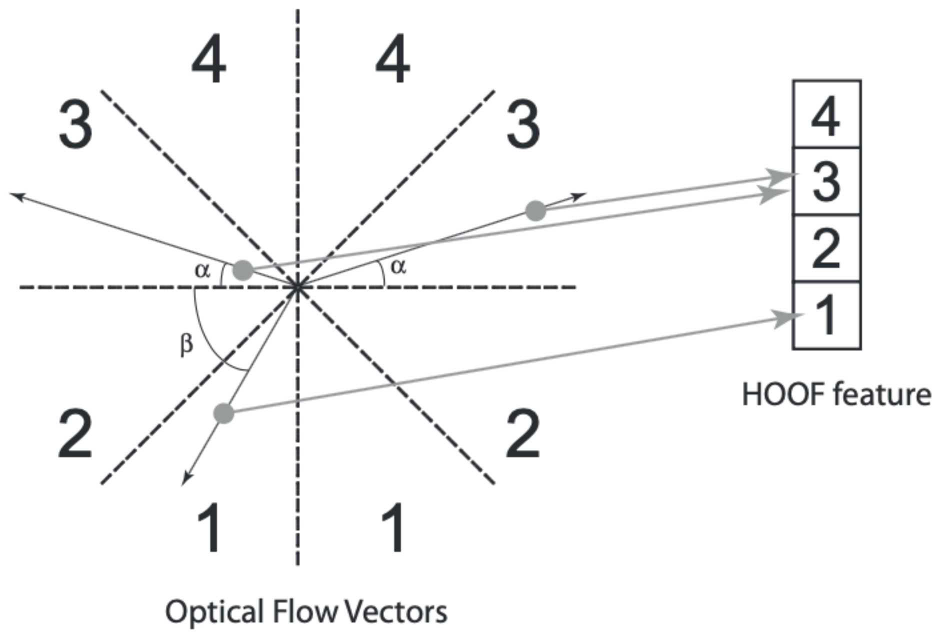


*Figure 1 Example HOOF for B=4. Figure taken from [1].*

You must use Lucas-Kanade method to calculate optical flow on each frame. You can use OpenCV built-in functions for this calculation. Always visualize optical flow results on an image similar to given example in figure 2.

*Figure 2 Example optical flow illustration. Figure taken from*[1]

You need to create HOOF for each frame in a given video, at the end of your operations on a video, you must have $T$ number of HOOF vectors($h_t$) with $B$ dimensions, where $T$ is the total number of frames. Since each video has different number of frames, we need a valid representation for any number of frames. To achieve this, you need to calculate the temporal mean of the HOOF's you calculated. Formal definition is given below where $\bar{h}$ is the temporal mean HOOF for a video in your dataset.

$$\bar{h} = \frac{1}{T}\sum_{t}^{T} h_t$$

$$\mathbf{h}_t = [\mathbf{h}_{t;1}, \mathbf{h}_{t;2}, \ldots, \mathbf{h}_{t;B}]^\top$$

After you have obtained temporal mean HOOFs for every training sample, you must apply PCA and reserve 90% of the variance. Next, you will be classifying the videos with k-NN, you must use Euclidean Distance between HOOFs. Your goal is to achieve the best performance by finding optimal $B$ and $k$. Keep your execution time reasonable. Compare the k-NN results with and without PCA applied in terms of accuracy.

**Important notes:**

- **Python 3.6 or higher is a must.**
- **Test your code before submitting, it must not crash.**
- **You may only use OpenCV, Numpy and Matplotlib.**
- **Your Assignment will be disregarded if your code does not work or a report is not submitted!**
- **Your Assignment will be disregarded if you use any other library than the ones stated above!**
- **READ THE DOCUMENT CAREFULLY.**

# Grading:

**Your grade will be penalized 10 out of 100 points for each minor fix before running the code regarding the assignment details.**
**30%: HOOF calculation**
**30%: PCA**
**10%: KNN**
**30%: Report**

# Submission Information:

Send all your source codes and report to the LMS. Your code should be clean and easy to read by possessing the following properties;

- *Clean structure:* The overall code should be neatly organized, where the related statements are grouped together with enough spacing among them.
- *Appropriate use of comments:* There should be comments explaining what the program, and different groups of statements are supposed to do. Don't overdo it.
- *Meaningful and consistent variable naming:* The names of variables should be meaningful with respect to the purpose and usage of these variables.

**Submission:** By uploading your code and report to LMS as a single ZIP archive. No other methods (e.g., by e-mail) accepted. (You may resubmit as many times as you want until the deadline).
**Warning:** DO NOT SHARE YOUR CODE WITH OTHERS. Your programs are checked and compared against each other using automated tools. Any act of cheating will be punished severely.
Also:

- Name your archive file uploaded exactly as requested. Your archive file must be named as **<NAME>_<SURNAME>_<STUDENTID>.zip.**
- Make sure that your program runs and gives the **expected output.**
- The first lines of your code must include your name, surname, student number, and department as a **comment**. An example comment is as follows:
  */* John Smith S0001 Department of Computer Science */*
- **Don't include your image and video files** in your archive

Good luck ☺

# References:

[1] Chaudhry, Rizwan & Ravichandran, Avinash & Hager, Gregory & Vidal, René. (2009). Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. 2012 IEEE Conference on Computer Vision and Pattern Recognition. 1932-1939. 10.1109/CVPRW.2009.5206821.