Introduction to Image and Video Processing
coronaproject 4: video processing
due May 28

Alexia Briassouli (alexia.briassouli@maastrichtuniversity.nl)

Spring 2020

Logistics:

Each of these mini coronaprojects will count for 25% of your final grade. They will be checked for plagiarism (software + text). There might be small bonuses in some places (it will say "**bonus**"). You are requested to please hand in:

1. A report with your answers to the assignment questions and figures with your results. It should be 5-6 pages, but this is just a general guideline. You can use any doc editor you like.
2. The code for producing these results *with clear comments in the code!*. You can use any programming language you are comfortable with (preferably Python or Matlab for this class, but others are welcome). You should explain what you think are important parts of the code in the report (e.g. if you use a special trick that you are proud of).

Your grade will depend on how clearly you present and explain your results in the report and code. You are allowed some freedom to explore, so there is no one correct answer. However, you should demonstrate you have understood the class material and how it applies to these projects.

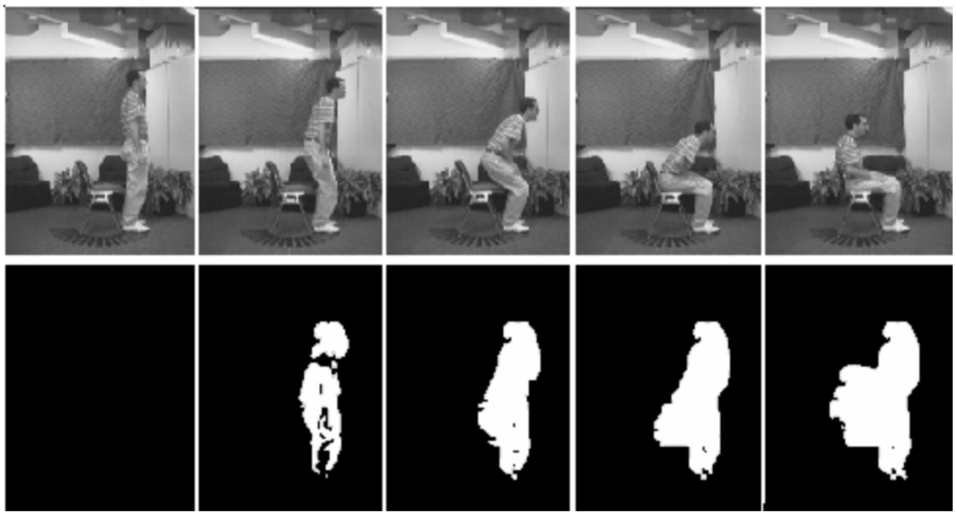
1 Motion Energy Images

Humans can easily and quickly recognize actions from low resolution information of the motion in a video. One way that has been used for representing videos of one activity recorded by a static camera by a binary template, the Motion Energy Image (MEI), or an equivalent grayscale one, the Motion History Image (MHI) (you can find additional information + images' source here: Ohio Sate uni project page on MEI, MHI with references).

1.1 Motion Energy Images

We consider part of the video, made up of w frames during which an activity takes place (like sit down, walk several steps...). **Motion Energy Images** are then defined as the pixel regions of the frame where there was motion:

$$MEI(x,y,t) = \begin{cases} 1 \text{ if there is motion in pixel}(x,y) \\ 0 \ else \end{cases} \tag{1}$$



2 Assignment on Motion Energy, Motion History Images

Choose **three** videos from **one** of the datasets below, e.g. walking-running-jogging from the KTH videos, or jump-bend-skip from the Weizmann videos. Use videos that are recorded from a static camera (the UCF ones might be more varied).

- KTH actions dataset
- Weizmann actions dataset
- UCF actions dataset

1. Finding the MEIs in the activities videos you chose (Matlab or Python). Make sure you choose w to be a reasonable temporal window (there is no one correct value for w). Display the resulting MEIs for two reasonable values of w.
   - In the definition of Eq. (1), I mention "if there is motion in pixel (x,y)". You can calculate the motion as follows:
     (a) Using inter-frame differences $D(x, y, t) = I(x, y, t) - I(x, y, t-1)$, for frame intensity equal to $I(x, y, t)$ at pixel (x, y) and frame t.
     (b) Using optical flow estimates, e.g. Lukas-Kanade, Farneback or other existing code for motion estimation.
   - **Bonus:**
     (a) Add noise to the image frames
     (b) Then find MEI, MHI (using inter-frame differences and/or optical flow)
     (c) Apply de-noising (you decide what kind) and find the MEI, MHI again. Display, and discuss/compare the results before and after de-noising.
2. Clean up the MEIs (binary images) with one or more morphological operations. Explain why you choose these operations. Display and discuss the results.
3. Find the outline of the MEI using a method of your liking, such as edge detection, morphological boundary extraction. Display it.
4. Extract the shape descriptor for the MEI outlines of the actions using Hu or Zernike moments, using ready-made functions. *You do not need to write this code from scratch, or explain it (as it's not the focus of this project), just use it as a tool to extract the shape descriptor.* Some suggestions for code - if you use them, make sure to refer to the author according to their license:

- Zernike matlab
- Zernike in Python
- Hu moments in Matlab
- OpenCV Hu moments in Python

5. Do a simple comparison of the shape descriptors you found between the three actions (e.g. by finding the Mean Squared Error or any other difference measure between the Hu descriptors of the three MEIs etc). Show and discuss your results.