Introduction to Image and Video Processing

Coronaproject 4: video processing

Selim Gilon* - i6192074

*Maastricht University - Department of Data Science and Knowledge Engineering

May 26^{th}, 2020

1 Motion Energy 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).

I chose 3 videos of 3 different actions from the KTH dataset. The actions are running, hand waving and walking.

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.

   For the running video, I chose $w = 3$ *because the motion from the action is pretty fast, the runner moves a lot in a short amount of time; therefore, we don't need a large temporal window to see the movement. If we want to be able to see the different "parts" of the runner's strides, we need to have a short w.*

   For the hand waving video, I chose $w = 10$ *because the motion from the action is not very fast. Therefore, in order to see a movement, we need a w large enough. If w=2 for example, there is almost no difference between the 2 frames because the arms haven't moved much.*

   For the walking video, I chose $w = 14$ *because the motion from the action is slow. Therefore, in order to see a nice movement, we need a w large enough.*

   - 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, t1), for frame intensity equal to I(x, y, t) at pixel (x, y) and frame t.
     The first experiment that I have done is finding the MEI's of a sequence of frames from a running video. This was done in order to calculate the motion over a complete stride. Here are the results:

Figure 1: Original frames (w=3) and MEIs

Then, I computed the MEIs for the 2 other actions and for those ones, I only used 2 frames.

Figure 2: Original frames and MEIs

Figure 3: Original frames and MEIs

(b) Using optical flow estimates, e.g. Lukas-Kanade, Farneback or other existing code for motion estimation.
I used the optical flow estimator Lukas-Kanade to estimate the motion. Here are the results for both axis for each actions.

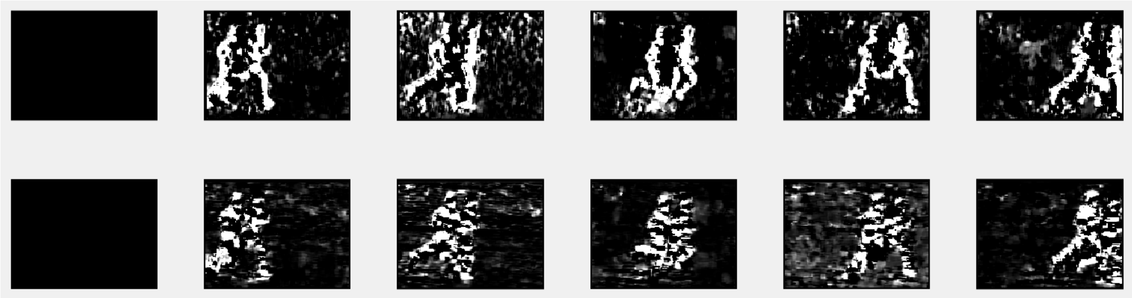

Figure 4: Result of Lukas-Kanade on running frames

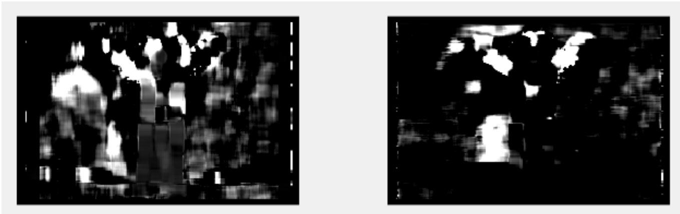

Figure 5: Result of Lukas-Kanade on hand waving frames

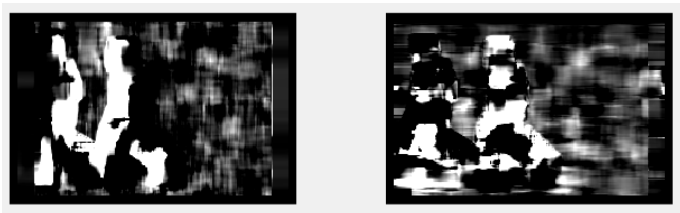

Figure 6: Result of Lukas-Kanade on walking frames

- Bonus:
  (a) Add noise to the image frames
  I added some noise to each frames of each action. A Gaussian noise was added to the running and hand waving frame while a Salt and pepper noise was added to the walking frames. You can see the noisy frames below:

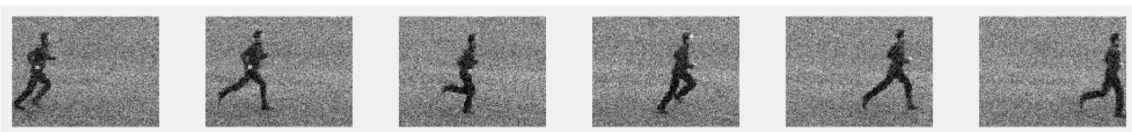

Figure 7: Noisy running frames

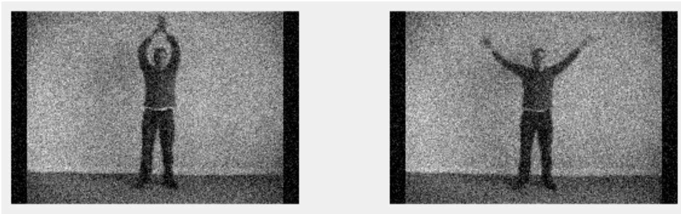

Figure 8: Noisy hand waving frames

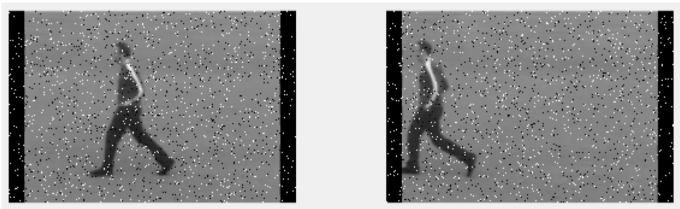

Figure 9: Noisy walking frames

(b) Then find MEI, MHI (using inter-frame differences and/or optical flow)
I computed the MEI using the inter-frame differences as described earlier. Here are the results:

Figure 10: MEI of noisy running frames

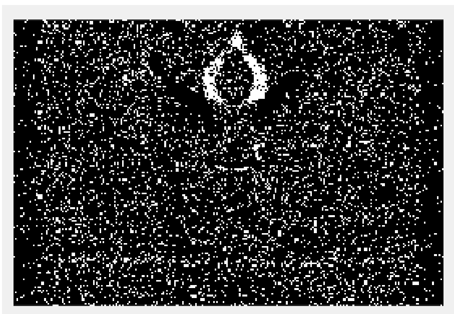

Figure 11: MEI of noisy hand waving frames

Figure 12: MEI of noisy walking frames

(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.
I then applied de-noising using a median filter. Here are the denoised frames:

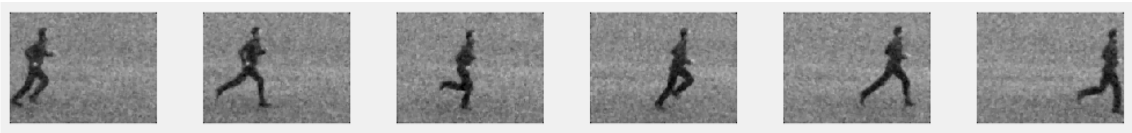

Figure 13: Denoised running frames

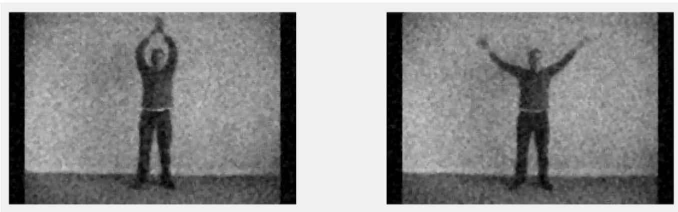

Figure 14: Denoised hand waving frames

Figure 15: Denoised walking frames

And here are te MEI of the denoised frames:

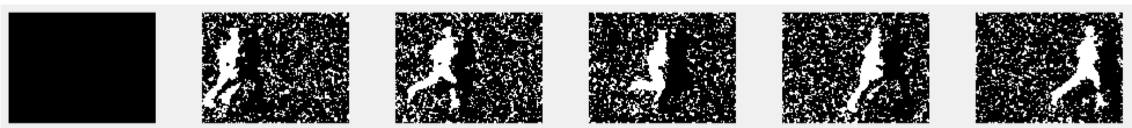

Figure 16: MEI of denoised running frames

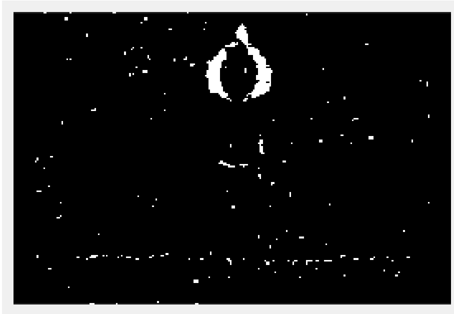

Figure 17: MEI of denoised hand waving frames

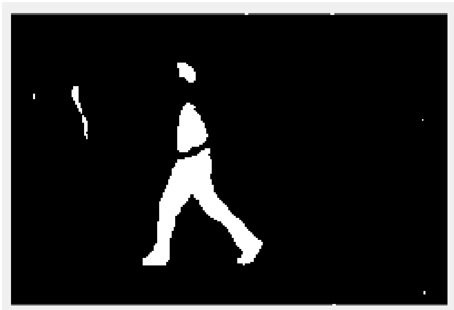

Figure 18: MEI of denoised walking frames

On the MEI of the frames with the salt and pepper noise (walk), we can see some white spots as well. The reason is the same as one described above.
After the denoising, we can see that there are less white spots (noise) on the MEIs than before the denoising process and the motion energy images are more clear. Almost all of the noisy white spots disappeared on the MEI of the denoised salt-pepper frames.*On the MEI of the frames with Gaussian noise, we can distinguish the last position and the new one (where the motion starts and ends). Indeed, the last position is white and the new one is almost completely black. There are a lot of white spot coming from the noise. Since the noise always follows the same distribution but is not of the same intensity a each corresponding pixels on pairs of frames, the difference of intensities that we compute to get the MEI results in some white spots. In other words: because the intensities of a pixel(x,y,t) and the pixel at the same location on the next frame (p(x,y,t+1)) can be very different from each other.*
On the MEI of the frames with the salt and pepper noise (walk), we can see some white spots as well. The reason is the same as one described above.
After the denoising, we can see that there are less white spots (noise) on the MEIs than before the denoising process and the motion energy images are more clear. Almost all of the noisy white spots disappeared on the MEI of the denoised saltpepper frames.

2. Clean up the MEIs (binary images) with one or more morphological operations. Explain why you choose these operations. Display and discuss the results.

   I would like to remove the small white spots which are noise (not from the human's motion). So, an erosion would be an option but I have noticed that the erosion deletes too many details so I decided to apply an opening with a disk shaped structural element of radius 1 or 2. This way, I get rid of the noise and keep the important details. Here are the results after the opening:

Figure 19: Opening on the running MEIs

Figure 20: Opening on the hand waving MEI

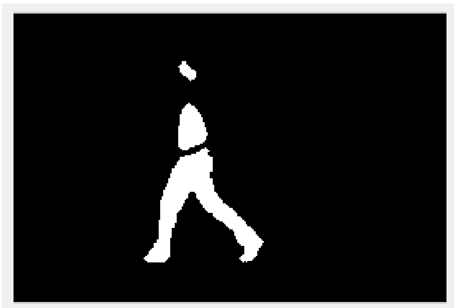

Figure 21: Opening on the walking MEI

We can clearly distinguish the shape from the human and there is no noise anymore: the only white areas on the MEIs are the from the motion of the human.

3. Find the outline of the MEI using a method of your liking, such as edge detection, morphological boundary extraction. Display it.

   I used the Canny method to extract the edges of the MEIs, here are the results for each action:

Figure 22: Edges detection on the running MEIs

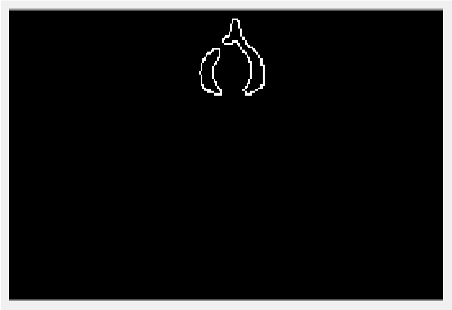

Figure 23: Edges detection on the hand waving MEI

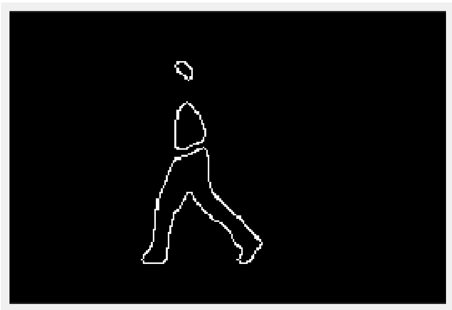

Figure 24: Edges detection on the walking MEI

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:

   I computed the Hu moments using the algorithm from Tejas K. Here are the Hu moments of the MEI outlines of each action:

   $hm = h1, h2, h3, h4, h5, h6, h7$

   $hu\_run1 = 2.1116, 2.7185, 2.2634, 1.2249, 2.0283, 1.8638, -0.8498, -0.3939$

   $hu\_run2 = 2.1983, 0.9390, 5.8271, 0.0273, 0.0074, 0.0211, -0.0038, -0.0082$

   $hu\_run3 = 2.4540, 3.1119, 1.9884, 0.2344, 0.1546, 0.4133, -0.1592, -0.0195$

   $hu\_run4 = 2.6655, 4.6637, 3.2541, 0.6437, 0.6091, 0.3916, 0.8747, -0.7857$

   $hu\_run5 = 2.5835, 1.4166, 11.4303, 0.2615, 0.1526, 0.1311, 0.2385, -0.1704$

   $hu\_hand = 1.3102, 0.0213, 0.5166, 0.0129, -0.0008, 0.0018, 0.0008, -0.0011$

   $hu\_walk = 3.0218, 2.8770, 21.3982, 1.9477, 12.2571, 3.1612, 2.2882, 0.4797$

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.

   In order to do a simple comparison between the shape descriptors of each action, I computed the mean squared error between the 3 pairs of actions.

   $mse(HumRun, HumHand) = 3.7290$

   $mse(HumWalk, HumHand) = 77.0711$

   $mse(HumRun, HumWalk) = 51.9988$

   $mse(humrun1, humrun2) = 3.2065$

$mse(humrun2, humrun3) = 2.4706$

we can see that the mean square error is very low between 2 Hu moments of a similar shape (from a same action) and this is logic: since the first 6 moments are invariant to translation, scale, rotation, and reflection and both set of moments are representing a human who is running, their moments' values are almost equal.

Comparing the Hu moments of different MEI outlines (from different actions), we observe that their mean square error can be pretty high and this is because they represent very dissimilar shapes.