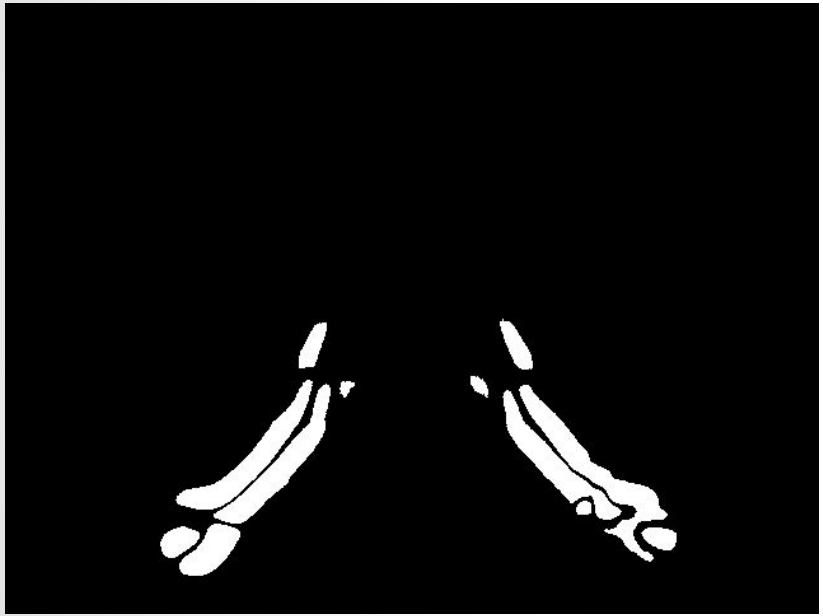
Computer Vision
Spring 2017
Problem Set #8

Yonathan Lim
yonathan@gatech.edu

1a: Binary image for frame 10

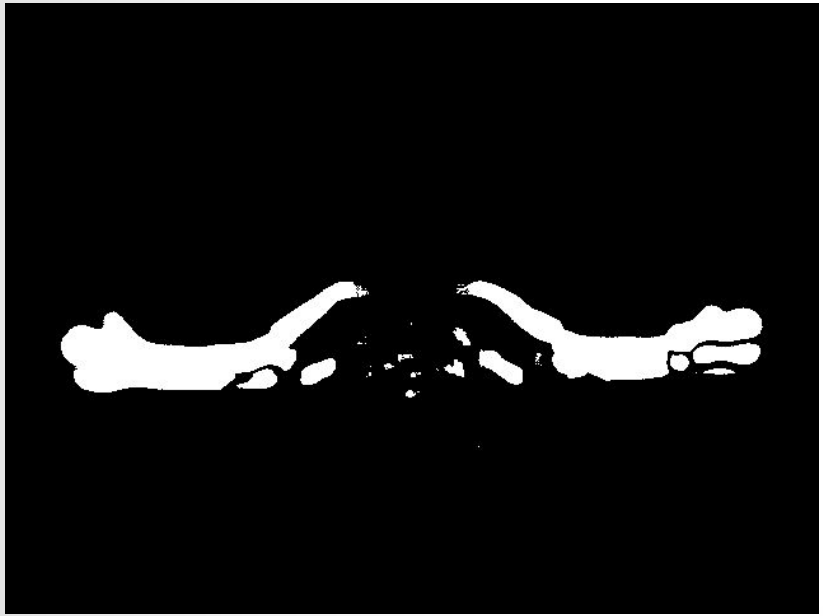
Binary image for frame 10 **- ps8-1-a-1.png**

1a: Binary image for frame 20



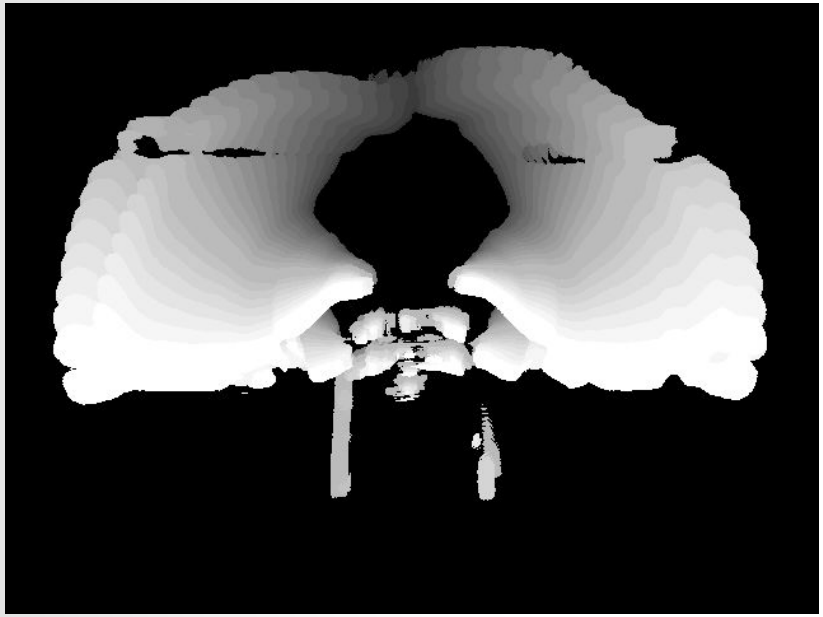
Binary image for frame 20 - **ps8-1-a-2.png**

1a: Binary image for frame 30



Binary image for frame 30 - **ps8-1-a-3.png**

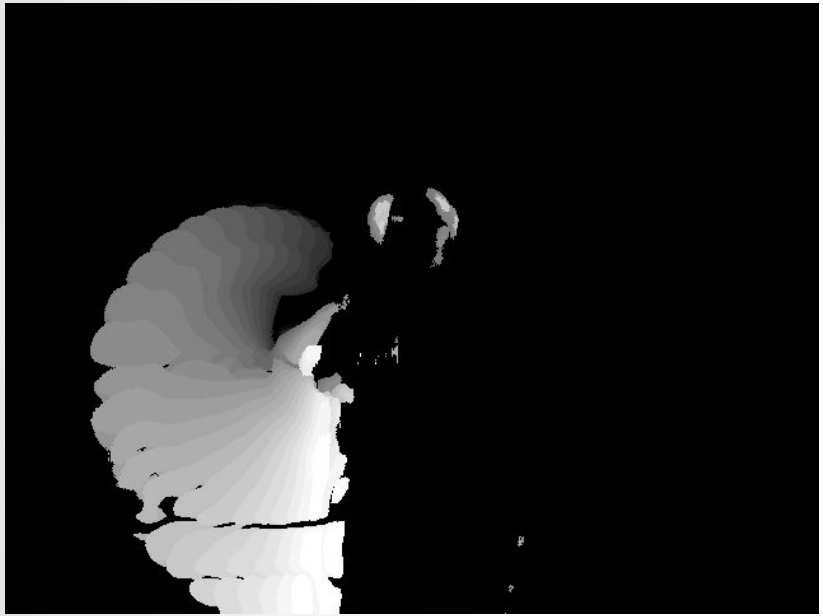
1b: MHI image for action A1



MHI image for action A1 - **ps8-1-b-1.png**

τ = ?

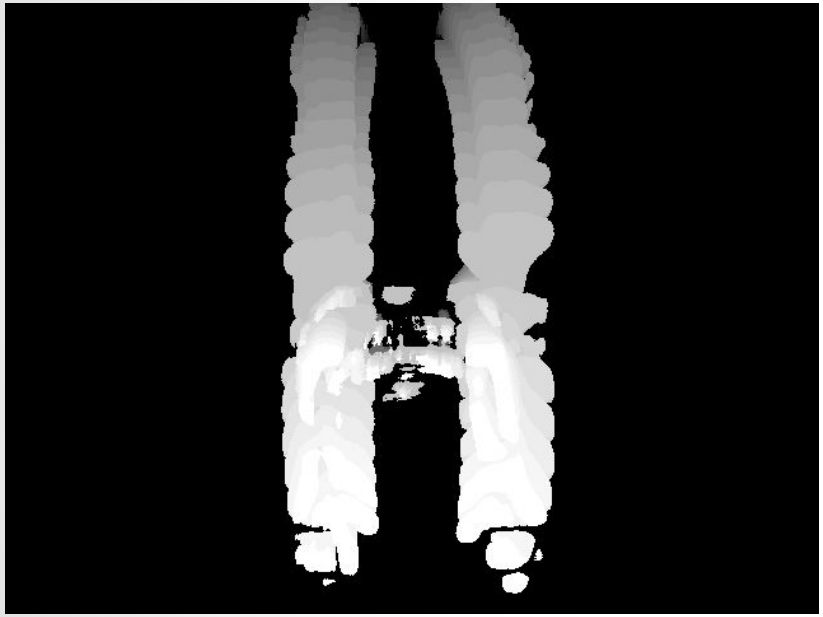
1b: MHI image for action A2



MHI image for action A2 - **ps8-1-b-2.png**

τ = ?

1b: MHI image for action A3



MHI image for action A3 - **ps8-1-b-3.png**

τ = ?

2a: The best confusion matrices you achieved

i) With unscaled central moments

1.	0.	0.
0.	1.	0.
0.	0.	1.

ii) With scaled central moments

1.	0.	0.
0.	1.	0.
0.	0.	1.

Description of any change made to the distance function (if required) to achieve this result:

- The distance function is euclidean distance (square root of the sum of the squares of the difference vector). Specific custom parameters (mhi_frame, theta, tau) are set to achieve this result.

2a: Custom Params

```
custom_params = {
    (1, 1, 1): dict(mhi_frame=109, theta=10, tau=35),
    (1, 1, 2): dict(mhi_frame=95, theta=10, tau=35),
    (1, 1, 3): dict(mhi_frame=111, theta=10, tau=35),
    (1, 2, 1): dict(mhi_frame=72, theta=10, tau=25),
    (1, 2, 2): dict(mhi_frame=64, theta=10, tau=20),
    (1, 2, 3): dict(mhi_frame=68, theta=10, tau=20),
    (1, 3, 1): dict(mhi_frame=84, theta=25, tau=25),
    (1, 3, 2): dict(mhi_frame=80, theta=25, tau=25),
    (1, 3, 3): dict(mhi_frame=77, theta=25, tau=25),

    (2, 1, 1): dict(mhi_frame=55, theta=10, tau=45),
    (2, 1, 2): dict(mhi_frame=55, theta=10, tau=45),
    (2, 1, 3): dict(mhi_frame=65, theta=10, tau=55),
    (2, 2, 1): dict(mhi_frame=50, theta=10, tau=50),
    (2, 2, 2): dict(mhi_frame=50, theta=10, tau=50),
    (2, 2, 3): dict(mhi_frame=50, theta=10, tau=50),
    (2, 3, 1): dict(mhi_frame=50, theta=10, tau=45),
    (2, 3, 2): dict(mhi_frame=50, theta=10, tau=45),
    (2, 3, 3): dict(mhi_frame=50, theta=10, tau=45),
    …
```

```
    …
    (3, 1, 1): dict(mhi_frame=97, theta=10, tau=45),
    (3, 1, 2): dict(mhi_frame=90, theta=10, tau=45),
    (3, 1, 3): dict(mhi_frame=92, theta=10, tau=45),
    (3, 2, 1): dict(mhi_frame=74, theta=10, tau=45),
    (3, 2, 2): dict(mhi_frame=80, theta=10, tau=45),
    (3, 2, 3): dict(mhi_frame=79, theta=10, tau=45),
    (3, 3, 1): dict(mhi_frame=75, theta=20, tau=60),
    (3, 3, 2): dict(mhi_frame=90, theta=20, tau=60),
    (3, 3, 3): dict(mhi_frame=85, theta=20, tau=60),
}
```

2b: The best confusion matrices you achieved

i) For P1

1.	0.	0.
0.	1.	0.
0.	0.	1.

ii) For P2

1.	0.	0.
0.	1.	0.
0.	0.	1.

ii) For P3

1.	0.	0.
0.	1.	0.
0.	0.	1.

Description of actions required to achieve this result:

- I saved all the motion history images to see the differences between them. Then, I modified the custom parameters to make sure that the motion history images aligned between each participant. I also changed the thresholds for each video so they were high enough to track only the motion and ignored the intensity change of the images that appear on some of the videos.

2b: Average of the confusion matrices

1.	0.	0.
0.	1.	0.
0.	0.	1.

Important Note

Please make sure your latest ps8.py and experiment.py are set to generate the images shown in your report. We will run your algorithms again locally using the same input videos to verify these results. We will not accept modifications to these files after the deadline if running your code fails.