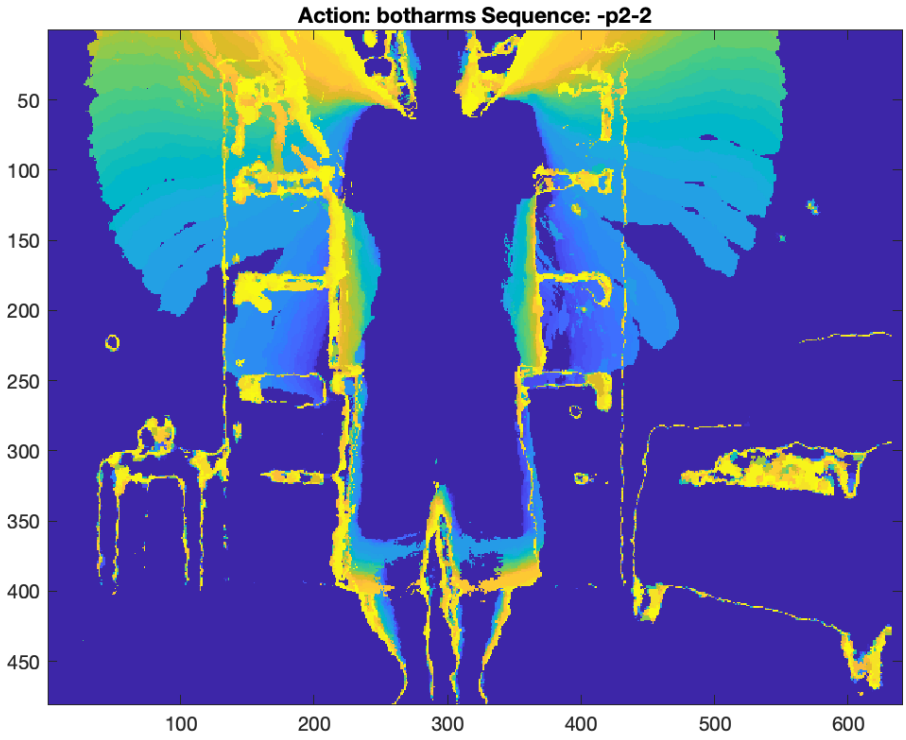
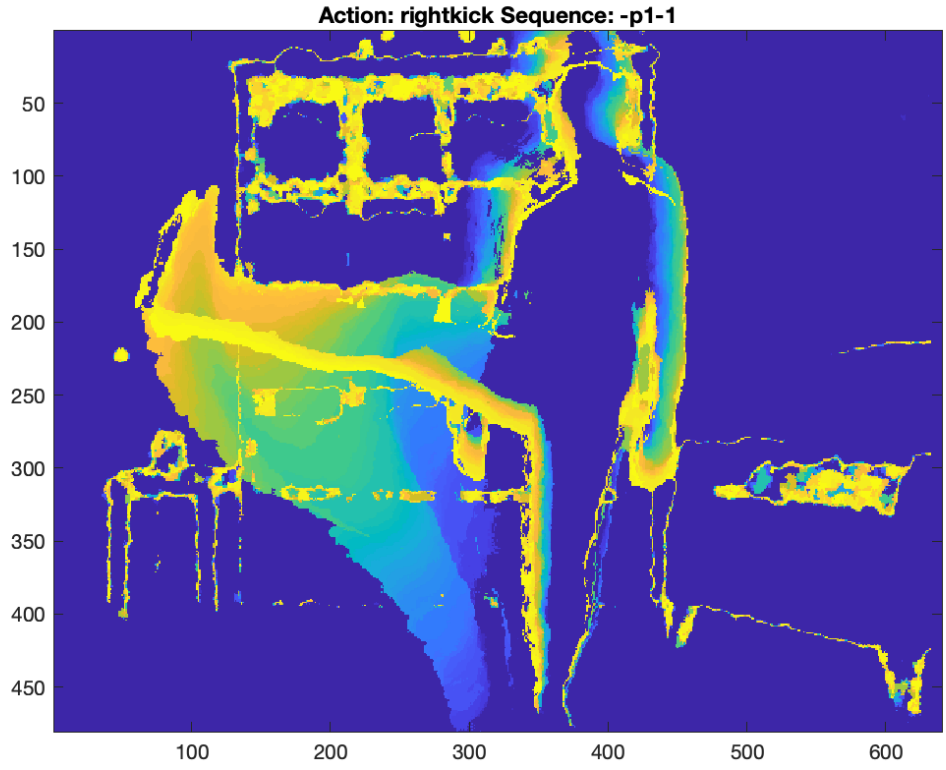
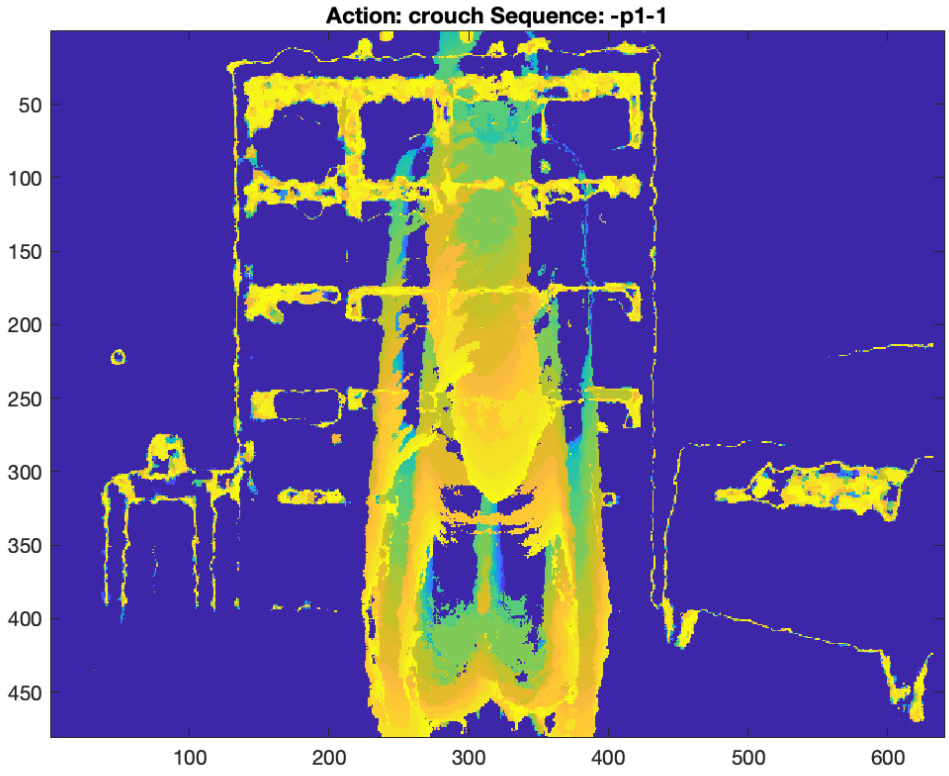
CS 6476
PS 5

1. Display three examples from different action categories .

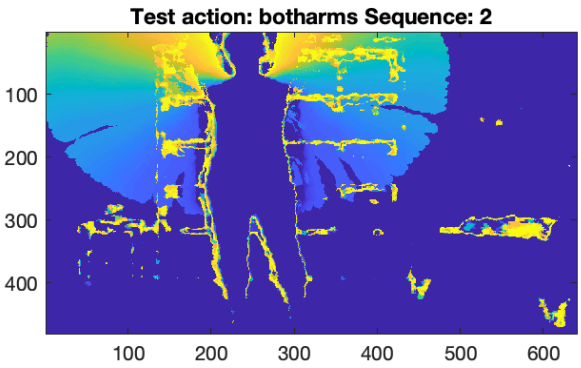
To extract foreground we subtract first frame from the second frame for two consecutive frames in the sequence folder. This method is introduced in class. It is very easy to implement. If a part of the body has moved, that area in the frame will have different values in the next frames, and similarly, the area which is originally empty will have new body part in it, which will change in pixel value. Either way, the absolute value will be bigger than stationary backgrounds. (Also we need to use threshold so that we don't introduce too much noise.) After that just apply the given formulas and obtain MHIs.

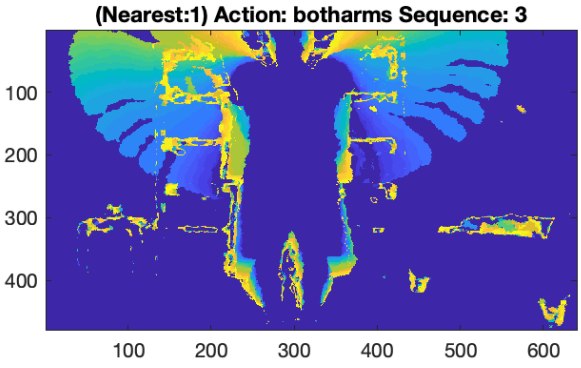


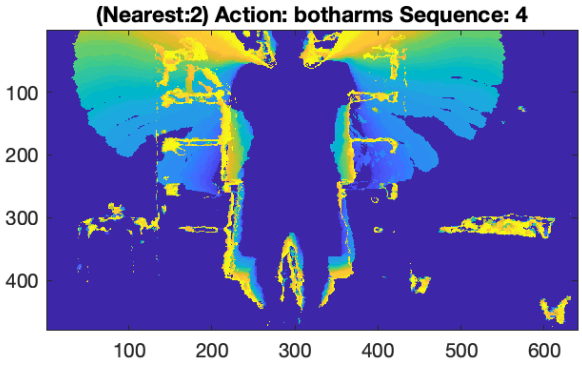

Action: rightkick Sequence: -p1-1
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600

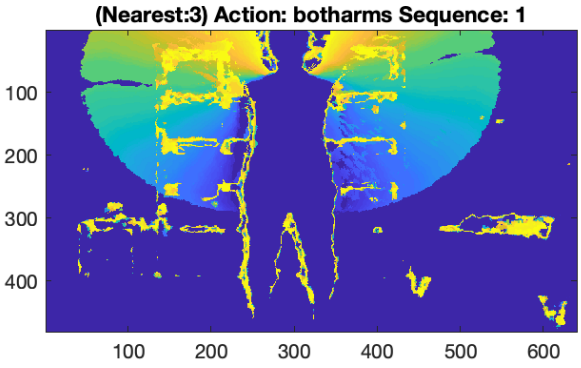

Action: crouch Sequence: -p1-1
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600

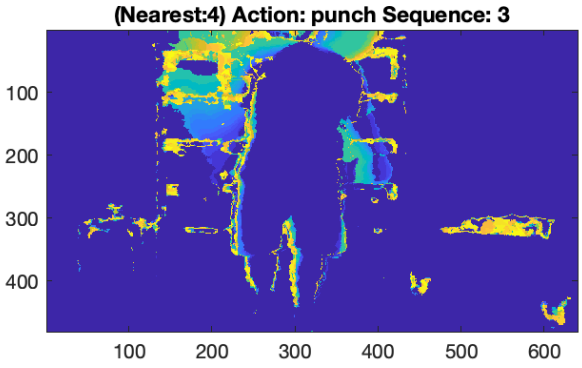
4. Display the results for two selected test examples (again, from different action categories), for K = 4. Apparently the nearest neighbor to one sequence is itself, and therefore I have excluded that to show the next 4 nearest sequences. Since there are only 4 sequences total in an action category, the 4th nearest sequence could be a sequence from another action.

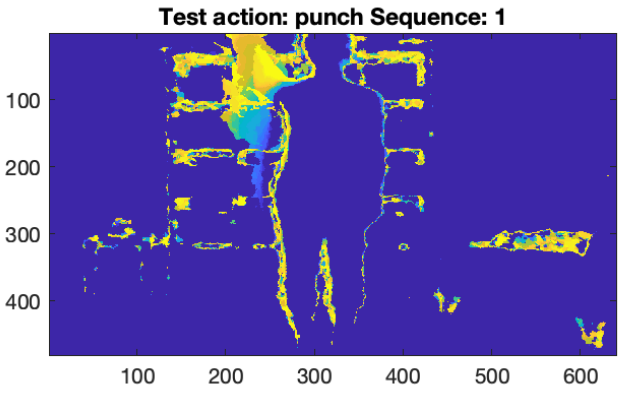


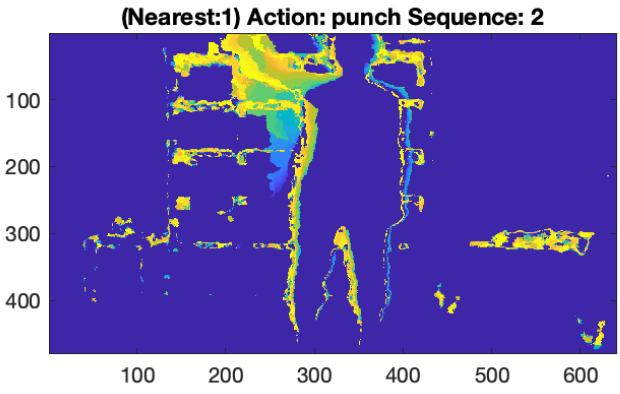


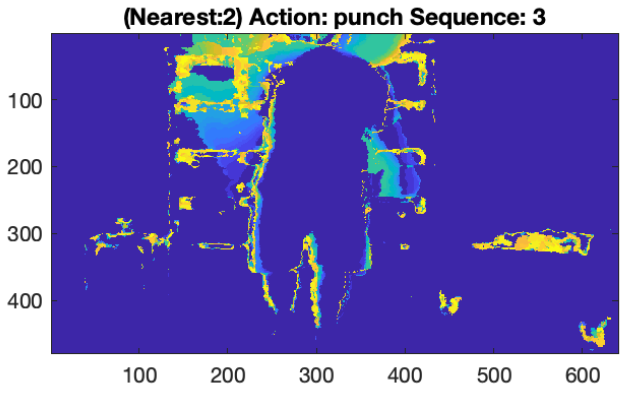


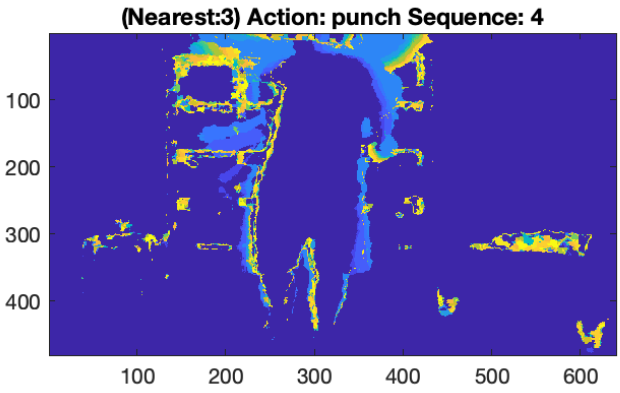


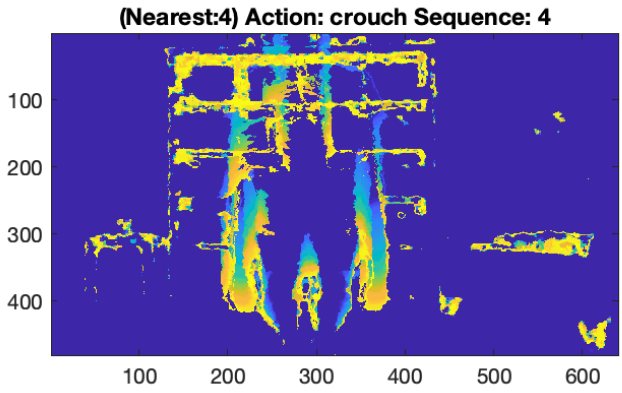












5.
Overall Recognition Rate: 75%
Recognition Rate per Class:
Botharms: 50%
Crouch: 100%
Leftarmup: 100%
Punch: 100%
Rightkick: 25%
Mean Recognition Rate per Class: 75%

Confusion Matrix:

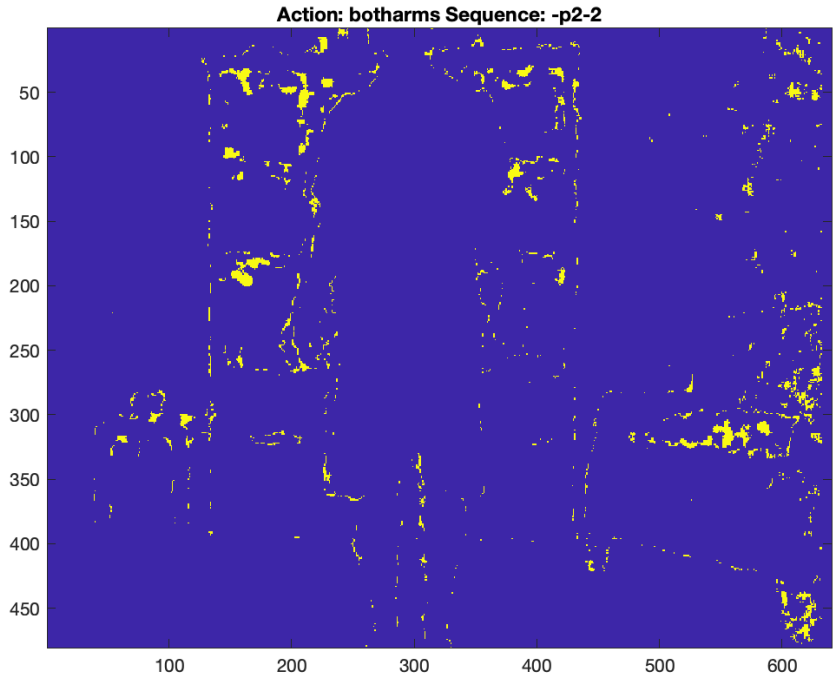
		Predicted Label				
		Botharms	Crouch	Leftarm Up	Punch	Rightkick
Actual Label	Botharms	2	0	0	0	2
	Crouch	0	4	0	0	0
	Leftarm up	0	0	4	0	0
	Punch	0	0	0	4	0
	Rightkick	2	1	0	0	1

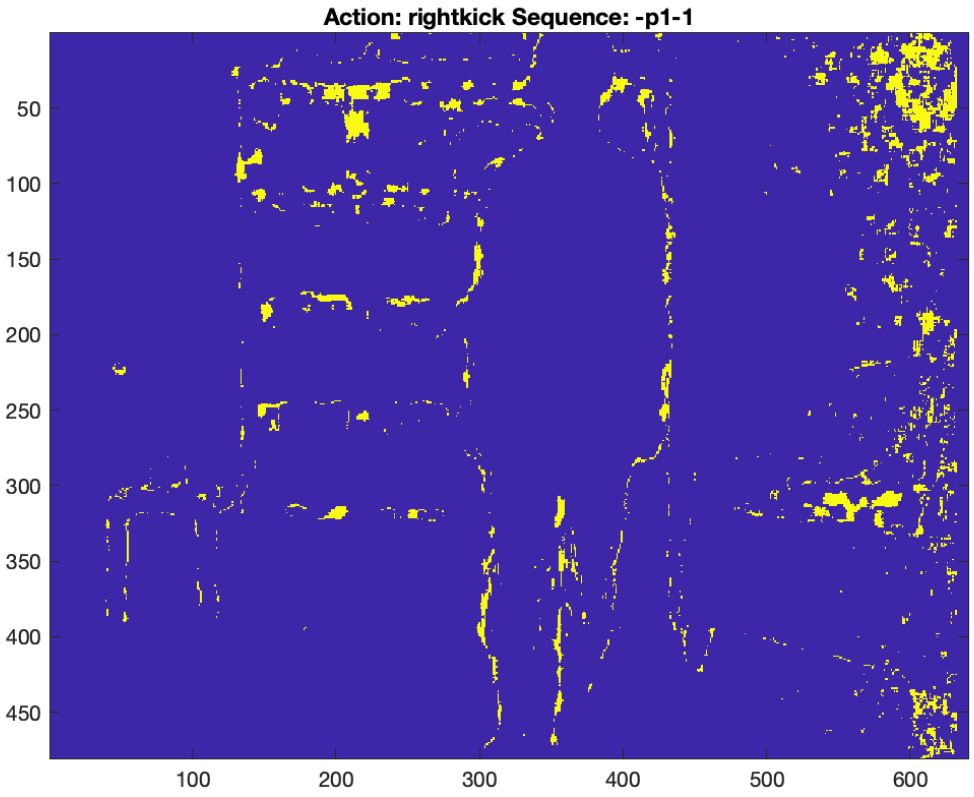
Recognition rate for 'botharms' is 50% and for 'rightkick' is 25%. The most confused class seems to be 'rightkick', which the algorithm predicted for 'botharms' and a 'crouch'. It might be because 'rightkick' itself is an overall body movement rather than other actions where only a part of the body moves, so the MHIs can be noisier. Also, if we look at MHIs for 'rightkick' they involve large area of movement similar to 'botharmup', despite different parts of the body. If we look at the Hu vectors of both actions, they do also seem very alike. It is possible that we didn't set good enough filter, causing noisy MHIs, and that Hu vectors didn't capture the big movements well enough to differentiate the exact locations of movement.

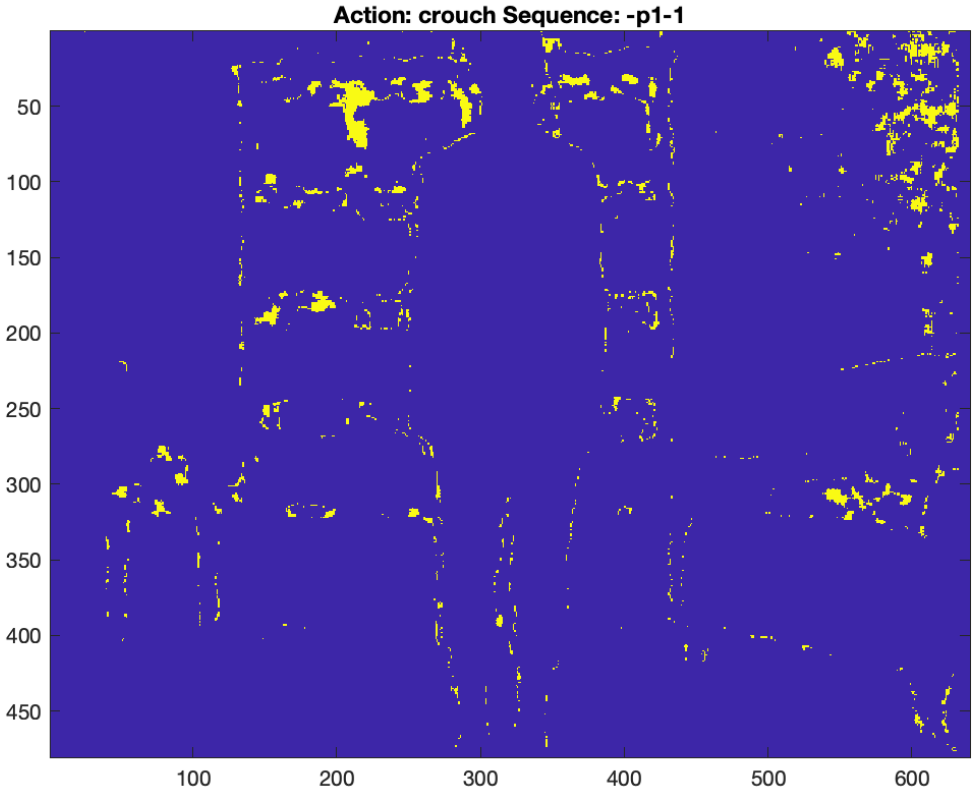
Extra Credit

1. My first approach is to use mean filter to extract foreground: for frame i, calculate the average values of previous (i-1) frames, and use |i – average values| > threshold to extract foreground. However, this method did successfully extract the silhouette of stationary parts, instead of capturing movements. (See folder 'Extra Credit/MHI_mean/' for MHI results and 'computeMHI_mean.mat' for code). Therefore, I did not proceed with this method.

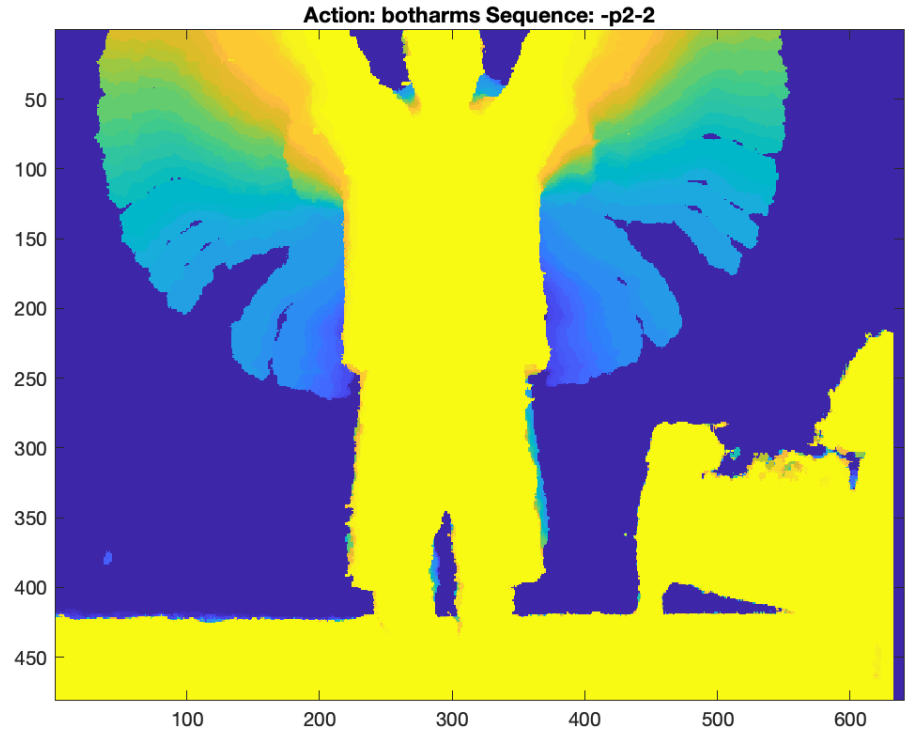
Same action sequences as in section 1:

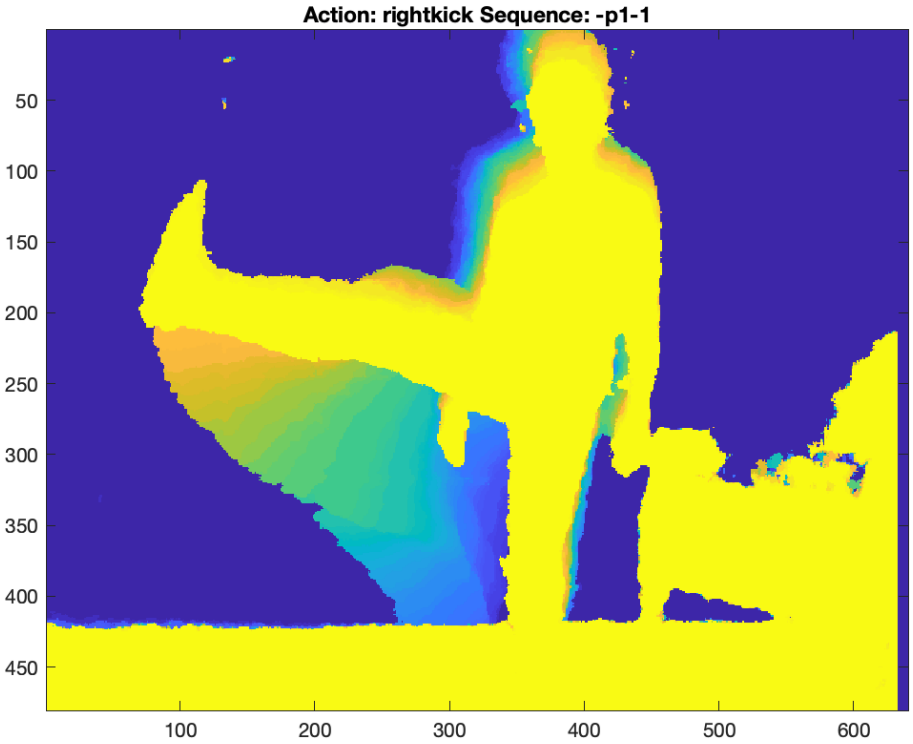


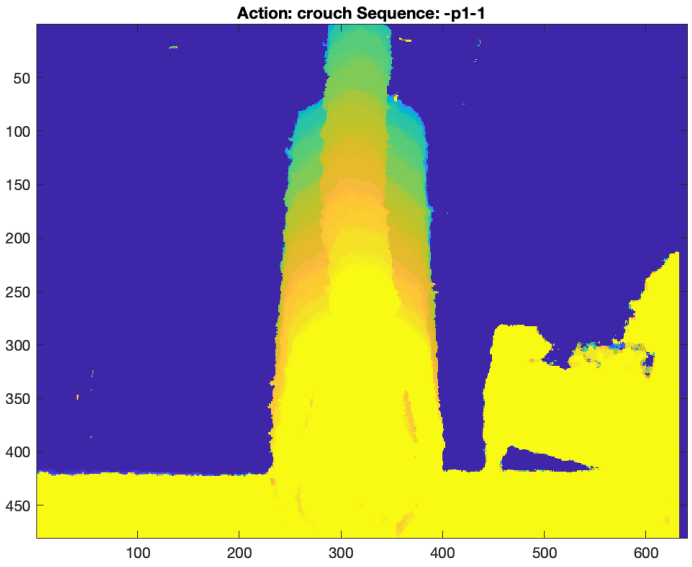

Action: rightkick Sequence: -p1-1
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600


Action: crouch Sequence: -p1-1
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600

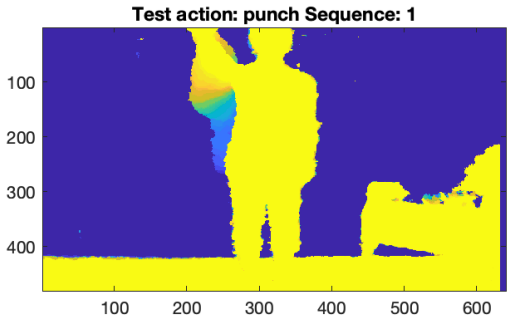
2. The second method I tried is the depth image. Since the given data is by itself a depth map, the foreground is simply the area where the values are super large. Here we use threshold 50000, any value bigger than 50000 is considered foreground. See folder 'Extra Credit/MHI_depth/' for MHI results and 'computeMHI_depth.mat' for code. As you can see this method extracts the foreground really well.

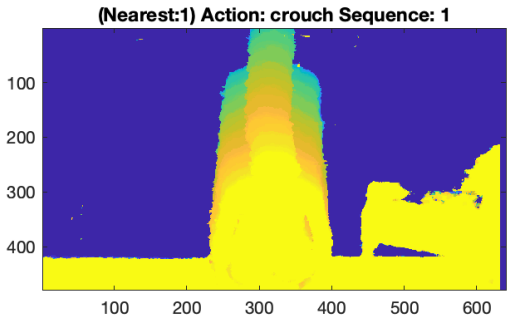


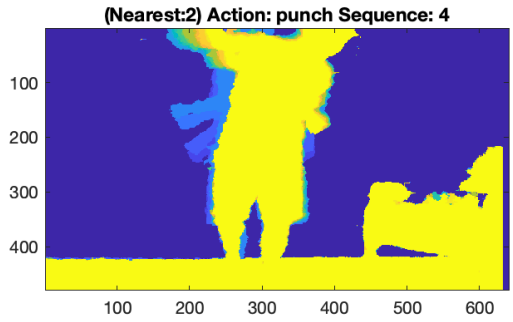


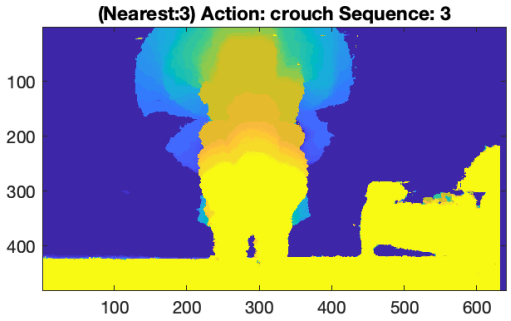


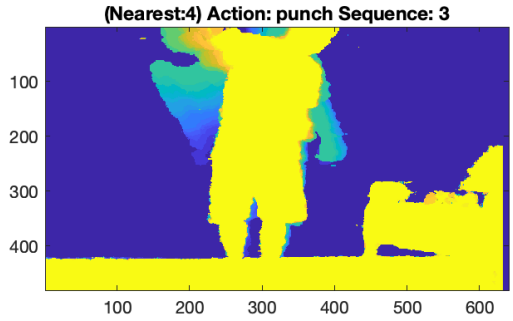
However, the matching results weren't as good as in section 1. Two crouch sequences were identified as nearest to punch.











And for leave-one-out cross validation, the result actually gets worse. Overall recognition rate becomes 55% and recognition rate for each class are:
Botharms up: 75%
Crouch: 75%
Leftarmup: 100%
Punch: 0%
Rightkick: 25%

Confusion matrix:

		Predicted Label				
		Botharms	Crouch	Leftarm Up	Punch	Rightkick
Actual Label	Botharms	3	0	0	0	1
	Crouch	0	3	1	0	0
	Leftarm up	0	0	4	0	0
	Punch	0	1	2	0	1
	Rightkick	1	1	1	0	1

The performance is definitely worse than in section 1. I am not sure why this is the case, since in my opinion the foreground extraction is better. And the algorithm is still confused by the rightkick action.