Practical Block III

Human Gesture Analysis

Juan Lao-Tebar

1. Introduction

In this practical work we use Dynamic Time Warping (DTW) [1] as a template matching algorithm, in order to find multiple instances of a given pattern in a sequence of different human gestures, indicating the beginning and ending point of each one. The dataset we use for training and testing is the Microsoft Research Cambridge-12 Kinect™ (MSRC-12)[1], that consists of sequences of human movements, represented as body-part locations in a 3D space, and the associated gesture to be recognized by the system.

The MSRC-12 dataset does not indicate the beginning of the gestures, just the end of them. Also, some gestures are not correctly labeled, and others look corrupted—the gesture has a glitch in the middle of the sequence, skipping or repeating some frames—. These issues not only complicate the training process, but also perverts the testing results and it is difficult to have a clear idea of how good is our approach.

In addition, the authors of the MSRC-12 dataset did not give clear instructions to the human volunteers who performed the recorded gestures; they only told them the *name* of the gesture—e.g, “shoot”—and the volunteers performed a gesture based on their subjective interpretation of the name. In this dataset you can find several patterns that describe the same gesture in a completely different way, especially for abstract gestures such as “Wind up the music”. For this reason we consider DTW a very bad approach for this problem, since it uses just one sample as a model and it is not able to learn the inherent probability distributions and correlations of the data.

In the code provided with this paper you will find two scripts that lets you perform any experiment using the best of our approaches.

- `demo2.m` is an interactive user-friendly script that lets you train models and test them against file sequences, as well as plotting error maps and visualizing animations of sequences, detections and more.
- `demo3.m` is a script to perform bulk automated tests and obtaining the result tables presented in this paper.

[1] https://www.microsoft.com/en-us/download/details.aspx?id=52283

2. Implementation and Experiments

In this section we describe the evolution of our approach, improving the system step by step, in order to increase the resulting F-Score and bringing the mean deviation of the beginning and ending points to 0, as much as possible.

The MSRC-12 dataset is composed by 594 sequence files. Each sequence performs a gesture of only one type, repeated multiple times. In our experiments we use 75% of the data for training the model and 25% for testing. Note that the MSRC-12 dataset is not correctly labeled, especially at the beginning and ending of each file. For this reason we discard all the frames before the first label and after the last label of each file, not only for the training process but also for the testing phase.

The training process consists on reading the training files—75% of the whole dataset—, extracting all the samples of the given gesture, choosing one sample as the *model sequence* and finally adjusting several parameters taking into account the rest of the samples.

The testing process consists on reading all the testing files—25% of the whole dataset—and then applying DTW to detect multiple instances of the *model sequence* using the adjusted parameters. For each positive case the system provides the beginning and ending points inside the testing file. After that we measure the accuracy of our model with two indicators:

- F_1 score, also including the precision and recall rates, to determine the quality of the model taking into account the true and false positives/negatives.
- Only for the true positive cases, we also calculate the mean deviation (MD) and mean absolute deviation (MAD) of the detected beginning and ending points compared to the real labels. These deviations indicate, in frames, the precision of our approach for detecting the exact position of the gesture in a sequence.

2.1. Normalization

The MSRC-12 dataset is not normalized: humans are not at the center of the space—point (0, 0, 0)—and it contains gestures from people with different statures.

We normalize each frame individually: first we set the *V* component of each body-part to 0, then we center the frame over the *hip center* and finally we multiply each body-part by a factor *f*.

$$f = \frac{c}{d}$$

Where *d* is the Euclidean distance between the *hip center* and the *spine* and *c* is a constant used for visualization purposes, that in our case is equal to 0.005—its value does not affect the system accuracy—. We normalize the scale of the entire body using the distance

between the *hip center* and the *spine* because, among all the possible distances between two different body-parts, this is the most invariant one[2].

The function defined in `normalizeFrame.m` performs the described process.

2.2. Approach #1: The Naive Approach

The first and most basic implementation of our algorithm consists on using an arbitrary sample as the *model sequence* and adjusting just one parameter: the DTW error threshold. In this approach we use as cost function the Euclidean distance between two different frames—80 dimensions—not only for matches but also for insertions and deletions.

At the training process, the DTW error threshold is computed just calculating the maximum DTW error obtained when we *compare*[3] individually the *model sequence* with the rest of the samples.

When we test the model against a file sequence, first we *search*[4] the *model sequence* inside the file sequence. After we obtain the DTW error map, the next step consists on determining the ending points of the gesture.

The detection of ending points is not trivial. If we recognize as ending points all the cells in the last row of the DTW error map with a value lower than the given DTW error threshold, we get a huge amount of false positives. Note that if a gesture is very similar to our *model sequence*, a large number of cells near—and not so near—the real ending point are considered possible endings, since the threshold is calculated using the worst-possible sample of the training set.

In this approach we consider as ending points all the cells in the last row of the error map that meet the following conditions:

1. The cell value is a local minimum in the sequence composed by the last row of the error map.
2. The cell value is lower than the computed DTW error threshold.

For calculating the beginning points we just perform a simple backtracking starting from the ending points, following the minimum value path, until we reach the first row of the error map.

Note that several ending points may end in the same beginning point. Due to this phenomena, we decided to test our algorithm in two different modes, as seen in figure 1.

---

[2] This property has been found with the script `findOptimalScale.m`.
[3] When we *compare* two sequences using DTW, we initialize the first row and the first column of the matrix with +\inf, and the first cell with 0.
[4] When we *search* a sequence A in a sequence B in DTW, we initialize the first column of the matrix with +\inf and the first row with 0, including the first cell.

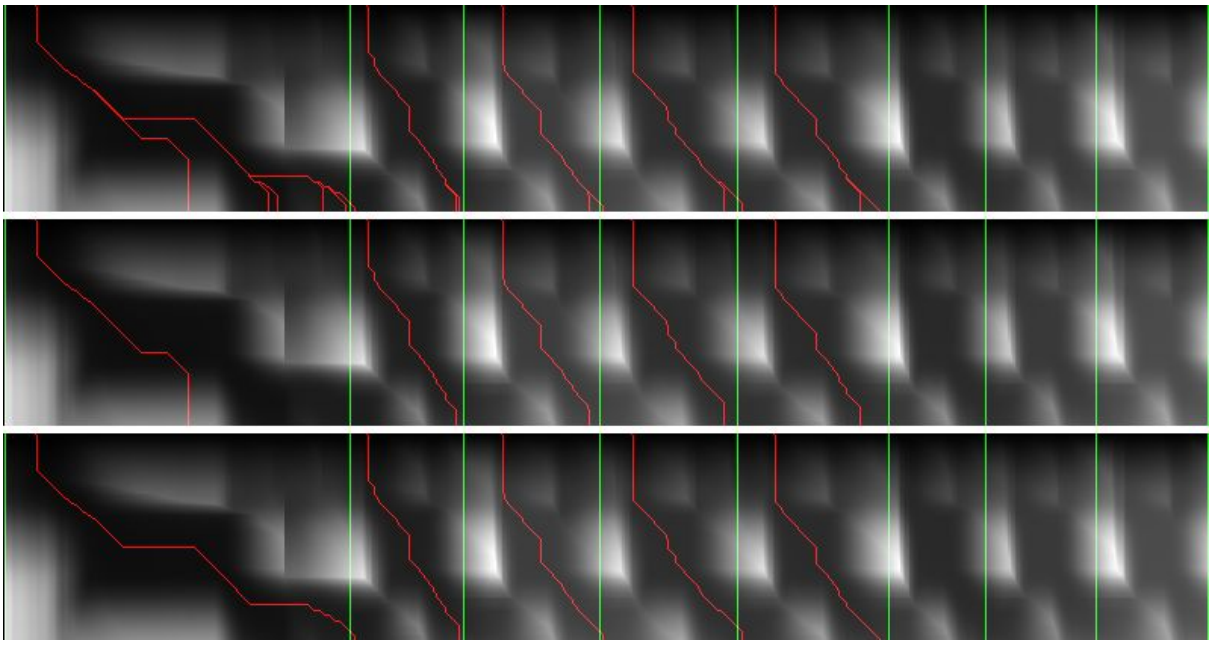

Figure 1. Detection of gestures in different modes. On the top image all the possible ramifications are shown. The other two images, from top to bottom, correspond to the *online* mode and the *offline* mode.

- *Online* mode: we only consider the first ending point starting from the left, and we discard the rest of the ramifications. This method offers a system with no lag, and it is recommended to be used in real-time scenarios, such as games.
- *Offline* mode: we only consider the last ending point starting from the left, and we discard the rest of the ramifications. This method is not suitable for real-time scenarios where the gesture must be detected instantaneously, but it offers a better overall accuracy—as it can be seen in the following experiments—.

The results obtained with this approach are presented in table 1 for the *online* mode and in 2 for the *offline* test.

2.3. Approach #2: Maximum Insertions Threshold

With the previous approach we get a huge number of false positives. We observed that a large amount of them are caused when the system decides to perform a lot of *insertions* in the DTW process. Figure 2 shows an example of this phenomena.

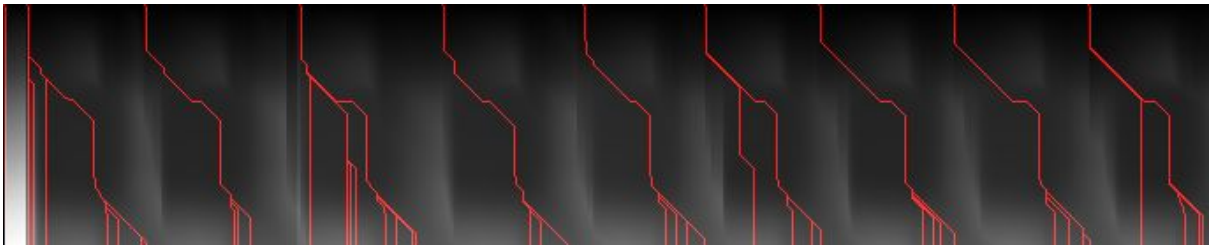

Figure 2. DTW error map for the detection of a gesture of type 1 in a file sequence of a gesture of type 6. Vertical segments represent *insertions* in the DTW process.

One possible way of avoid these cases consists on defining a *maximum insertions threshold*. This new variable, during the training process, takes as value the maximum number of insertions performed by any sample. During the testing process, when we detect and ending point and we backtrack it, if the number of insertions is greater than this threshold, we discard the ending point.

Figure 3 show what happens with the previous scenario after applying this change.

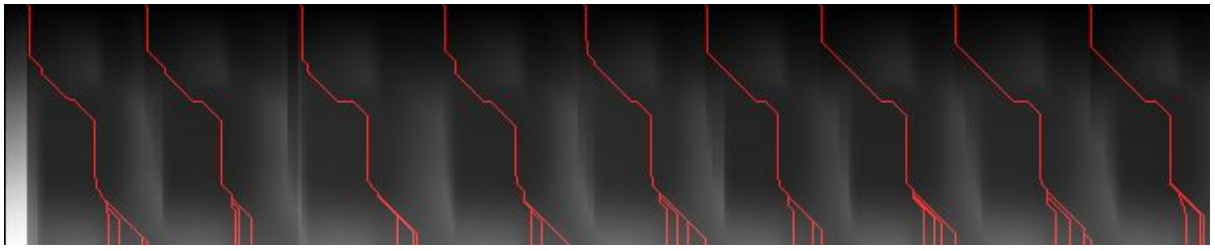

Figure 3. Same context as figure 2, but applying the maximum insertions threshold.

Tables 3 and 4 show the results after applying this change.

2.4. Approach #3: Maximum Last Insertion Threshold

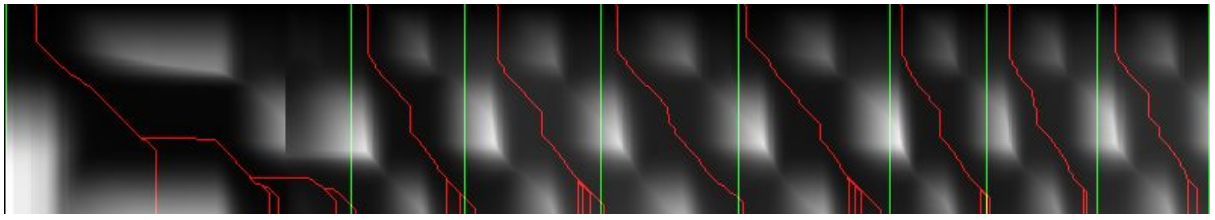

Figure 4. False positives caused by early detections, due to an abuse of the *insertion* feature in last steps of the DTW process.

As figure 4 shows, we also have a lot of false positives at the end of each detected gesture that abuse of the *insertion* feature in the DTW process.

To avoid this problem we introduce the *maximum last insertion threshold*, that is analogous to the *maximum insertions threshold*, but with a difference: we only take into account the insertions that end up in the candidate ending point; the rest of the insertions are ignored. Figure 5 shows a test example of this approach.

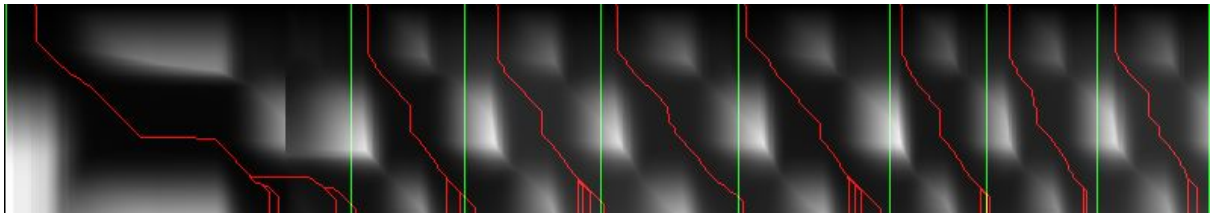

Figure 5. Same context as figure 5, but applying the last insertion threshold.

Tables 5 and 6 show the results after applying this change.

2.5. Approach #4: Regularization

As said in the previous sections, the volunteers who performed the human gestures for the MSRC-12 dataset did completely different kind of movement patterns for the same gesture type. When we adjust our thresholds with just 1 model sequence, we are calculating a threshold that tries to keep all those different kind of movements at the same level. The problem is that the higher recall rate we get, the poorer precision we obtain.

One regularization method we designed for our system consists on representing the computed data (DTW error and number of insertions) as random variables that follow a normal distribution. Then we just set a threshold to recognize 97.5% of the true positives, discarding outliers, thus decreasing the number of false positives, thus increasing the precision of the system.

Tables 7 and 8 show the results after setting our thresholds in the mean value plus 2 standard deviations of the computed variables.

2.6. Approach #5: Weighted Cost Function

As a last improvement for our system, we modify the cost function to add different weight to each body-part depending on how relevant it is in the *model sequence*.

For our approach, instead of assigning an arbitrary relevance to each body-part, we prefered to calculate it dynamically.

Tables 9 and 10 show the results obtained when the relevance of a body-part is linearly proportional to the distance traveled during the *model sequence*:

$$w_i = \frac{d_i}{max(d)}$$

Tables 11 and 12 show the results obtained when the relevance of a body-part is logarithmically proportional to the distance traveled during the *model sequence*:

$$w_i = \frac{\log(d_i)}{\log(max(d))}$$

Tables 13 and 14 show the results obtained when the relevance of a body-part is linear and inversely proportional to the distance traveled during the *model sequence*:

$$w_i = 1 - \frac{d_i}{max(d)}$$

The function defined in `calculateWeights.m` calculates the weight vector for the body-parts for any given sequence.

2.7. Using a Different Sample as *Model Sequence*

In the previous experiments we used the first sample of the given gesture as *model sequence*. Tables 15 and 16 show the results when train the system with the 50th sample; and tables 17 and 18 show the results when we train it with the 100th sample.

Note that in scripts `demo2.m` and `demo3.m` you can use any sample you wish as *model sequence*.

3. Discussion

As it can be seen in tables 19 and 20, the maximum F-Score we obtain in our best approach using Dynamic Time Warping with several improvements is 28.45%, while other authors (Bloom, Argyriou & Makris, 2013) [2] obtain F-Scores of 64.3% using AdaBoost,[3] 74.7% with Dynamic Feature Selection and 76.5% with Random Forest.[4]

Approach	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
#1	4.40%	99.64%	8.42%	8.34	14.22	-31.58	57.42
#2	11.95%	89.16%	20.09%	9.89	15.37	-27.80	42.06
#3	12.89%	83.49%	21.20%	9.19	14.43	-22.16	33.55
#4	18.33%	75.41%	26.78%	8.57	13.89	-18.01	26.51
#5	22.02%	67.47%	28.32%	7.43	12.51	-14.25	21.24

Table 19. Summarized mean results of our approaches in *online* mode.

Approach	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
#1	4.40%	99.64%	8.42%	8.34	14.22	-4.05	10.55
#2	11.95%	89.16%	20.09%	9.89	15.37	-5.37	12.95
#3	12.89%	83.49%	21.20%	9.19	14.43	-4.46	12.09
#4	18.33%	75.41%	26.78%	8.57	13.89	-18.01	26.51
#5	22.02%	67.47%	28.32%	7.43	12.51	-2.98	9.76

Table 19. Summarized mean results of our approaches in *offline* mode.

Our improvements increased the mean F-Score while slightly augmenting the precision rate and heavily decreasing the recall rate at the same time. In other words, we detect less false positives at the cost of losing a considerable amount of true positives.

Regarding to the detection of the beginning and ending point of the true positive cases, DTW worked really well. Using the first sample as *model sequence*, In *online*

mode—recommended for real-time applications such as video-games—we detect the gesture 14.25 frames *before* it ends (0.47 seconds approximately); and we detect the beginning of the gesture 7.43 frames after it starts. In the *offline* mode, we get even better results: the ending point is detected only 2.98 frames *before* the gesture ends (0.1 seconds approximately).

When we tried different samples as *model sequence*, we observed that the result does not vary significantly.

4. Conclusions and Future Work

In this practical work we used Dynamic Time Warping as a template matching algorithm, in order to find multiple instances of a given pattern in a sequence of different human gestures, indicating the beginning and ending point of each one. The dataset we used for training and testing is the Microsoft Research Cambridge-12 Kinect™, that consists of sequences of human movements, represented as body-part locations in a 3D space, and the associated gesture to be recognized by the system.

We conclude that it is not a good idea to approach this problem using DTW. As commented in the introduction, Dynamic Time Warping is not a machine learning technique; it does not learn the probabilistic distributions of the data. However, if we continue through this path, we propose two improvements that may increase the F-Score.

The first improvement consists on finding the optimal regularization parameter (i.e. threshold to consider a sample as *outlier*) using machine learning techniques. We got good results recognizing 97.5% of the true positives, but we could not try other threshold values due to time reasons.

The second improvement we propose consists on adjusting the weights of the cost function using also a machine learning technique. In our approach we set them inversely proportional to the distance they travel in the *model sequence*, but it is an arbitrary reason without fundament.

5. References

[1] Berndt, D. J., & Clifford, J. (1994, July). Using dynamic time warping to find patterns in time series. In *KDD workshop* (Vol. 10, No. 16, pp. 359-370).

[2] Bloom, V., Argyriou, V., & Makris, D. (2013, October). Dynamic Feature Selection for Online Action Recognition. In *HBU* (pp. 64-76).

[3] Schapire, R. E., & Freund, Y. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci, 55*(1), 119-139.

[4] Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

6. Appendix: Results

#1: The Naive Approach

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	4.10%	100.00%	7.87%	21.45	22.65	-29.78	40.67
2	4.93%	100.00%	9.40%	-2.88	13.56	-46.41	91.01
3	3.37%	100.00%	6.52%	12.51	16.36	-25.39	57.66
4	6.19%	100.00%	11.66%	11.93	16.56	-37.57	52.89
5	4.11%	97.67%	7.89%	6.30	11.52	-2.08	19.80
6	4.61%	97.96%	8.81%	9.01	11.85	-32.73	48.82
7	3.50%	100.00%	6.77%	4.36	18.70	-7.73	21.63
8	4.94%	100.00%	9.42%	9.04	10.63	-27.99	49.23
9	4.54%	100.00%	8.68%	6.40	13.81	-45.81	65.63
10	3.95%	100.00%	7.61%	5.30	13.57	-33.36	68.42
11	5.10%	100.00%	9.71%	6.96	10.30	-69.08	111.63
12	3.48%	100.00%	6.72%	9.72	11.11	-21.06	61.62
Mean	4.40%	99.64%	8.42%	8.34	14.22	-31.58	57.42

Table 1. Results of the Naive Approach in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	4.10%	100.00%	7.87%	21.45	22.65	1.66	6.04
2	4.93%	100.00%	9.40%	-2.88	13.56	7.68	11.82
3	3.37%	100.00%	6.52%	12.51	16.36	-7.89	14.34
4	6.19%	100.00%	11.66%	11.93	16.56	-3.98	5.30
5	4.11%	97.67%	7.89%	6.30	11.52	1.33	6.05
6	4.61%	97.96%	8.81%	9.01	11.85	-6.41	15.22
7	3.50%	100.00%	6.77%	4.36	18.70	-2.63	9.15
8	4.94%	100.00%	9.42%	9.04	10.63	0.54	4.59
9	4.54%	100.00%	8.68%	6.40	13.81	-15.44	16.24
10	3.95%	100.00%	7.61%	5.30	13.57	-15.17	15.63
11	5.10%	100.00%	9.71%	6.96	10.30	-8.53	13.91
12	3.48%	100.00%	6.72%	9.72	11.11	0.21	8.33
Mean	4.40%	99.64%	8.42%	8.34	14.22	-4.05	10.55

Table 2. Results of the Naive Approach in *offline* mode.

#2: Maximum Insertions Threshold

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	8.60%	99.01%	15.82%	24.79	25.49	-35.44	43.04
2	11.43%	99.12%	20.49%	-1.96	14.38	-44.73	69.87
3	5.13%	100.00%	9.76%	22.11	25.28	-29.86	70.11
4	11.65%	82.41%	20.41%	8.58	13.18	-16.10	16.12
5	12.59%	79.65%	21.75%	6.70	15.56	-3.42	19.22
6	6.56%	97.96%	12.30%	9.87	12.66	-35.00	46.35
7	24.73%	38.98%	30.26%	5.92	12.14	-9.25	15.60
8	11.62%	100.00%	20.81%	9.30	14.67	-17.70	27.09
9	11.45%	98.15%	20.50%	7.31	14.66	-27.98	30.56
10	16.11%	87.83%	27.22%	5.60	12.16	-30.33	42.61
11	10.10%	95.80%	18.27%	7.14	10.42	-60.62	96.79
12	13.45%	90.99%	23.43%	13.35	13.87	-23.20	27.32
Mean	11.95%	89.16%	20.09%	9.89	15.37	-27.80	42.06

Table 3. Results after applying the maximum insertions threshold in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	8.60%	99.01%	15.82%	24.79	25.49	-5.24	12.49
2	11.43%	99.12%	20.49%	-1.96	14.38	1.02	17.16
3	5.13%	100.00%	9.76%	22.11	25.28	-10.69	23.96
4	11.65%	82.41%	20.41%	8.58	13.18	-3.56	4.41
5	12.59%	79.65%	21.75%	6.70	15.56	0.63	9.73
6	6.56%	97.96%	12.30%	9.87	12.66	-3.68	15.62
7	24.73%	38.98%	30.26%	5.92	12.14	-4.42	11.43
8	11.62%	100.00%	20.81%	9.30	14.67	-0.56	5.27
9	11.45%	98.15%	20.50%	7.31	14.66	-15.09	15.89
10	16.11%	87.83%	27.22%	5.60	12.16	-15.46	15.76
11	10.10%	95.80%	18.27%	7.14	10.42	-7.71	13.08
12	13.45%	90.99%	23.43%	13.35	13.87	0.28	10.55
Mean	11.95%	89.16%	20.09%	9.89	15.37	-5.37	12.95

Table 4. Results after applying the maximum insertions threshold in *offline* mode.

#3: Maximum Last Insertion Threshold

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	9.12%	99.01%	16.69%	24.79	25.49	-28.06	30.12
2	13.22%	97.35%	23.28%	-0.72	14.68	-37.50	51.06
3	5.35%	100.00%	10.17%	22.16	25.31	-29.67	68.43
4	12.02%	73.15%	20.65%	6.95	10.97	-6.59	6.70
5	12.59%	79.65%	21.75%	6.70	15.56	-3.42	19.22
6	6.57%	97.96%	12.31%	9.87	12.66	-34.34	46.38
7	24.71%	35.59%	29.17%	5.25	11.47	-8.57	13.82
8	13.50%	100.00%	23.79%	9.52	15.77	-14.41	17.45
9	13.19%	68.52%	22.12%	2.73	8.84	-10.57	11.13
10	18.64%	76.52%	29.98%	2.91	8.55	-16.75	17.50
11	10.38%	95.80%	18.73%	10.90	14.08	-57.45	99.23
12	15.40%	78.38%	25.74%	9.26	9.78	-18.63	21.59
Mean	12.89%	83.49%	21.20%	9.19	14.43	-22.16	33.55

Table 5. Results after applying the maximum last insertion threshold in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	9.12%	99.01%	16.69%	24.79	25.49	-5.24	12.49
2	13.22%	97.35%	23.28%	-0.72	14.68	2.94	14.78
3	5.35%	100.00%	10.17%	22.16	25.31	-10.69	23.96
4	12.02%	73.15%	20.65%	6.95	10.97	-3.16	4.01
5	12.59%	79.65%	21.75%	6.70	15.56	0.63	9.73
6	6.57%	97.96%	12.31%	9.87	12.66	-3.68	15.62
7	24.71%	35.59%	29.17%	5.25	11.47	-4.76	10.56
8	13.50%	100.00%	23.79%	9.52	15.77	-3.17	7.88
9	13.19%	68.52%	22.12%	2.73	8.84	-8.72	9.52
10	18.64%	76.52%	29.98%	2.91	8.55	-11.32	11.62
11	10.38%	95.80%	18.73%	10.90	14.08	-4.60	15.84
12	15.40%	78.38%	25.74%	9.26	9.78	-1.74	9.04
Mean	12.89%	83.49%	21.20%	9.19	14.43	-4.46	12.09

Table 6. Results after applying the maximum last insertion threshold in *offline* mode.

#4: Regularization

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	9.59%	92.08%	17.37%	34.24	36.42	-17.82	26.02
2	11.30%	100.00%	20.31%	-0.33	15.62	-37.08	53.61
3	38.78%	63.33%	48.10%	9.95	11.57	-11.85	11.85
4	16.09%	76.85%	26.60%	7.34	11.44	-7.64	7.82
5	12.07%	87.21%	21.20%	6.48	14.59	-6.65	22.45
6	7.63%	95.92%	14.14%	10.47	13.12	-29.96	41.65
7	28.57%	22.03%	24.88%	1.92	6.37	-5.64	7.10
8	20.38%	92.97%	33.43%	16.02	21.08	-8.58	12.94
9	15.91%	58.33%	25.00%	0.79	6.90	-9.37	9.50
10	25.41%	67.83%	36.97%	1.46	6.71	-13.45	13.52
11	12.13%	90.76%	21.41%	6.86	14.77	-56.16	97.71
12	22.15%	57.66%	32.00%	7.67	8.08	-11.88	13.90
Mean	18.33%	75.41%	26.78%	8.57	13.89	-18.01	26.51

Table 7. Results after regularizing in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	9.59%	92.08%	17.37%	34.24	36.42	-0.94	14.47
2	11.30%	100.00%	20.31%	-0.33	15.62	3.35	15.57
3	38.78%	63.33%	48.10%	9.95	11.57	-6.23	6.23
4	16.09%	76.85%	26.60%	7.34	11.44	-3.44	4.55
5	12.07%	87.21%	21.20%	6.48	14.59	-0.84	10.88
6	7.63%	95.92%	14.14%	10.47	13.12	-6.35	14.96
7	28.57%	22.03%	24.88%	1.92	6.37	-3.73	5.66
8	20.38%	92.97%	33.43%	16.02	21.08	-0.80	9.57
9	15.91%	58.33%	25.00%	0.79	6.90	-7.73	8.10
10	25.41%	67.83%	36.97%	1.46	6.71	-9.63	9.92
11	12.13%	90.76%	21.41%	6.86	14.77	-8.18	19.43
12	22.15%	57.66%	32.00%	7.67	8.08	0.03	6.64
Mean	18.33%	75.41%	26.78%	8.57	13.89	-3.71	10.50

Table 8. Results after regularizing in *offline* mode.

#5: Weighted Cost Function

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	17.12%	88.12%	28.66%	22.09	22.52	-2.90	4.92
2	7.35%	99.12%	13.68%	-5.73	16.67	-38.68	78.40
3	13.14%	81.67%	22.63%	5.99	13.14	-18.59	20.86
4	19.70%	84.26%	31.93%	9.23	12.47	-4.72	5.07
5	10.82%	95.93%	19.45%	7.25	15.75	-13.09	29.41
6	6.94%	96.94%	12.96%	8.71	11.73	-30.70	38.70
7	18.33%	18.64%	18.49%	2.15	4.32	-4.09	5.62
8	12.57%	91.41%	22.10%	11.26	13.84	-12.59	13.60
9	28.30%	69.44%	40.21%	3.62	10.40	-9.27	11.53
10	17.22%	89.57%	28.89%	3.48	12.03	-22.30	32.86
11	12.60%	94.96%	22.24%	7.21	12.94	-54.11	88.88
12	9.62%	82.88%	17.24%	10.42	11.05	-16.93	22.64
Mean	14.48%	82.74%	23.21%	7.14	13.07	-19.00	29.37

Table 9. Results after using a linearly weighted cost function in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	17.12%	88.12%	28.66%	22.09	22.52	4.60	6.62
2	7.35%	99.12%	13.68%	-5.73	16.67	1.70	13.56
3	13.14%	81.67%	22.63%	5.99	13.14	-9.01	9.03
4	19.70%	84.26%	31.93%	9.23	12.47	-2.56	3.61
5	10.82%	95.93%	19.45%	7.25	15.75	-2.57	5.55
6	6.94%	96.94%	12.96%	8.71	11.73	-6.86	13.82
7	18.33%	18.64%	18.49%	2.15	4.32	-2.17	4.17
8	12.57%	91.41%	22.10%	11.26	13.84	2.72	5.36
9	28.30%	69.44%	40.21%	3.62	10.40	-7.27	11.08
10	17.22%	89.57%	28.89%	3.48	12.03	-9.88	11.62
11	12.60%	94.96%	22.24%	7.21	12.94	-1.73	12.18
12	9.62%	82.88%	17.24%	10.42	11.05	6.85	9.80
Mean	14.48%	82.74%	23.21%	7.14	13.07	-2.18	8.87

Table 10. Results after using a linearly weighted cost function in *offline* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	17.12%	88.12%	28.66%	21.91	22.35	-2.94	4.96
2	8.23%	100.00%	15.21%	-3.04	14.69	-38.47	59.34
3	13.21%	81.67%	22.74%	6.00	13.15	-18.59	20.86
4	19.44%	83.33%	31.52%	9.28	12.43	-4.68	5.03
5	10.84%	95.93%	19.48%	7.20	15.69	-13.45	29.78
6	6.89%	96.94%	12.87%	9.10	12.12	-30.70	38.70
7	18.49%	18.64%	18.57%	2.15	4.32	-4.09	5.62
8	12.49%	89.84%	21.93%	11.13	13.70	-12.41	13.41
9	28.11%	64.81%	39.22%	3.47	9.77	-8.60	10.75
10	17.88%	89.57%	29.81%	3.47	12.04	-21.81	32.89
11	11.86%	94.96%	21.08%	7.18	12.96	-54.15	88.92
12	11.92%	79.28%	20.73%	10.77	11.39	-17.59	19.61
Mean	14.71%	81.92%	23.48%	7.39	12.88	-18.96	27.49

Table 11. Results after using a logarithmically weighted cost function in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	17.12%	88.12%	28.66%	21.91	22.35	4.60	6.62
2	8.23%	100.00%	15.21%	-3.04	14.69	5.45	11.98
3	13.21%	81.67%	22.74%	6.00	13.15	-9.02	9.03
4	19.44%	83.33%	31.52%	9.28	12.43	-2.54	3.59
5	10.84%	95.93%	19.48%	7.20	15.69	-2.57	5.55
6	6.89%	96.94%	12.87%	9.10	12.12	-6.84	13.84
7	18.49%	18.64%	18.57%	2.15	4.32	-2.17	4.17
8	12.49%	89.84%	21.93%	11.13	13.70	2.25	5.27
9	28.11%	64.81%	39.22%	3.47	9.77	-6.57	10.39
10	17.88%	89.57%	29.81%	3.47	12.04	-9.71	11.66
11	11.86%	94.96%	21.08%	7.18	12.96	-1.98	12.47
12	11.92%	79.28%	20.73%	10.77	11.39	5.68	9.17
Mean	14.71%	81.92%	23.48%	7.39	12.88	-1.95	8.65

Table 12. Results after using a logarithmically weighted cost function in *offline* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	13.84%	78.22%	23.51%	28.48	29.09	-9.80	20.08
2	13.75%	97.35%	24.10%	-3.51	13.99	-24.35	26.40
3	52.70%	32.50%	40.21%	6.50	6.50	-5.48	5.48
4	14.42%	73.15%	24.09%	5.60	10.16	-8.30	8.50
5	13.31%	83.14%	22.95%	6.99	14.77	-3.72	24.43
6	8.12%	93.88%	14.95%	9.20	13.27	-34.52	51.66
7	38.55%	27.12%	31.84%	5.39	6.95	-4.46	4.51
8	25.05%	89.84%	39.18%	14.20	23.41	-4.56	14.48
9	19.29%	65.74%	29.83%	2.81	8.09	-9.94	11.24
10	31.18%	50.43%	38.54%	0.16	8.75	-7.03	8.12
11	12.46%	72.27%	21.26%	7.37	8.80	-50.92	70.08
12	21.61%	45.95%	29.39%	5.96	6.36	-7.90	9.86
Mean	22.02%	67.47%	28.32%	7.43	12.51	-14.25	21.24

Table 13. Results after using a linear inversely weighted cost function in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	13.84%	78.22%	23.51%	28.48	29.09	2.77	12.50
2	13.75%	97.35%	24.10%	-3.51	13.99	3.25	15.55
3	52.70%	32.50%	40.21%	6.50	6.50	-3.20	3.20
4	14.42%	73.15%	24.09%	5.60	10.16	-3.53	4.16
5	13.31%	83.14%	22.95%	6.99	14.77	-0.21	14.55
6	8.12%	93.88%	14.95%	9.20	13.27	-8.04	17.88
7	38.55%	27.12%	31.84%	5.39	6.95	-1.63	2.12
8	25.05%	89.84%	39.18%	14.20	23.41	-0.05	11.27
9	19.29%	65.74%	29.83%	2.81	8.09	-7.14	9.12
10	31.18%	50.43%	38.54%	0.16	8.75	-4.86	6.20
11	12.46%	72.27%	21.26%	7.37	8.80	-9.43	14.54
12	21.61%	45.95%	29.39%	5.96	6.36	-3.75	6.02
Mean	22.02%	67.47%	28.32%	7.43	12.51	-2.98	9.76

Table 14. Results after using a linear inversely weighted cost function in *offline* mode.

Using a Different Sample as *Model Sequence*

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	8.53%	28.71%	13.15%	-1.47	23.82	-27.36	27.36
2	35.90%	12.39%	18.42%	2.28	2.28	-2.29	3.69
3	48.98%	40.00%	44.04%	2.84	9.88	-5.36	5.43
4	60.71%	15.74%	25.00%	0.31	1.25	-1.26	1.26
5	20.66%	91.28%	33.69%	3.65	24.15	-16.06	17.09
6	11.18%	76.53%	19.51%	12.23	12.40	-14.23	19.54
7	20.55%	12.71%	15.71%	2.88	2.92	-0.98	1.12
8	34.75%	64.06%	45.05%	5.36	7.97	-3.41	4.35
9	18.29%	13.89%	15.79%	2.40	2.40	-2.69	3.16
10	18.85%	68.70%	29.59%	8.82	23.88	-4.14	10.70
11	22.68%	59.66%	32.87%	4.81	6.29	-33.33	37.73
12	52.63%	45.05%	48.54%	7.46	7.46	-5.95	5.98
Mean	29.48%	44.06%	28.45%	4.30	10.39	-9.76	11.45

Table 15. Results after using the 50th sample as *model sequence* in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	8.53%	28.71%	13.15%	-1.47	23.82	-25.70	25.70
2	35.90%	12.39%	18.42%	2.28	2.28	-2.12	3.69
3	48.98%	40.00%	44.04%	2.84	9.88	-3.23	3.67
4	60.71%	15.74%	25.00%	0.31	1.25	-0.76	0.76
5	20.66%	91.28%	33.69%	3.65	24.15	-13.80	13.97
6	11.18%	76.53%	19.51%	12.23	12.40	-5.16	13.69
7	20.55%	12.71%	15.71%	2.88	2.92	-0.44	0.85
8	34.75%	64.06%	45.05%	5.36	7.97	-1.57	2.76
9	18.29%	13.89%	15.79%	2.40	2.40	-2.44	3.19
10	18.85%	68.70%	29.59%	8.82	23.88	2.68	9.93
11	22.68%	59.66%	32.87%	4.81	6.29	-20.49	21.85
12	52.63%	45.05%	48.54%	7.46	7.46	-3.35	3.68
Mean	29.48%	44.06%	28.45%	4.30	10.39	-6.37	8.64

Table 16. Results after using the 50th sample as *model sequence* in *offline* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	13.51%	24.75%	17.48%	8.03	11.67	-10.19	12.50
2	39.53%	15.04%	21.79%	6.12	6.12	-6.46	6.48
3	44.78%	25.00%	32.09%	7.49	7.69	-3.46	3.46
4	13.46%	53.70%	21.52%	12.13	13.87	-3.78	3.89
5	19.97%	86.05%	32.42%	10.58	26.53	-11.52	13.65
6	11.95%	85.71%	20.97%	10.47	13.16	-15.95	19.40
7	23.08%	7.63%	11.46%	1.69	1.69	-1.71	1.80
8	39.18%	52.34%	44.82%	3.82	7.16	-0.21	2.59
9	30.95%	36.11%	33.33%	7.01	7.06	-4.92	5.40
10	19.44%	6.09%	9.27%	0.99	1.06	-1.51	1.51
11	12.95%	72.27%	21.97%	4.93	7.07	-50.81	74.79
12	37.16%	61.26%	46.26%	4.22	11.87	-17.50	18.07
Mean	25.50%	43.83%	26.12%	6.46	9.58	-10.67	13.63

Table 17. Results after using the 100th sample as *model sequence* in *online* mode.

Gesture	Precision	Recall	F-Score	Begin MD	Begin MAD	End MD	End MAD
1	13.51%	24.75%	17.48%	8.03	11.67	-7.20	9.61
2	39.53%	15.04%	21.79%	6.12	6.12	-6.19	6.40
3	44.78%	25.00%	32.09%	7.49	7.69	-2.11	2.11
4	13.46%	53.70%	21.52%	12.13	13.87	-3.12	3.25
5	19.97%	86.05%	32.42%	10.58	26.53	-8.90	11.28
6	11.95%	85.71%	20.97%	10.47	13.16	-3.13	13.03
7	23.08%	7.63%	11.46%	1.69	1.69	-1.61	1.69
8	39.18%	52.34%	44.82%	3.82	7.16	0.92	2.64
9	30.95%	36.11%	33.33%	7.01	7.06	-4.20	4.72
10	19.44%	6.09%	9.27%	0.99	1.06	-1.40	1.40
11	12.95%	72.27%	21.97%	4.93	7.07	-16.47	18.34
12	37.16%	61.26%	46.26%	4.22	11.87	-9.05	11.11
Mean	25.50%	43.83%	26.12%	6.46	9.58	-5.20	7.13

Table 18. Results after using the 100th sample as *model sequence* in *offline* mode.