5
Detecting Missed and Anomalous segments in Long-Term Actions

In the last chapter, we discussed a Hidden Markov Model-based approach to assess the pace of performers with mid-level skills. Such performers perform the entire long-term sequence correctly but tend to take an incorrect pace while performing posture transitions. Amateurs or first time performers, on the other hand tend to forget action steps and perform some unwanted action movements while performing a long term action sequence. The HMM-based approach discussed in the previous chapter learns the action transition probabilities based on training data which are recordings of expert performances. Such a model can decode sequences that follow the same pose transitions as experts and fail to decode sequences that do not follow the same order. In this chapter, we propose a framework for analyzing and issuing feedback reports of action segments that were missed or anomalously performed. This involves comparing the performed sequence with the standard action sequence and notifying when misalignments occur. We propose an exemplar based Approximate String Matching(ASM) technique for detecting such anomalous and missing segments in action sequences. We compare the results with those obtained from the conventional Dynamic Time Warping (DTW) algorithm for sequence alignment. It is seen that the alignment of the action sequences under conventional DTW fails in the presence of missed action segments and anomalous segments due to its boundary condition constraints. The performance of the two techniques has been tested on a second version of Sun Salutation dataset that now includes sequences of amateur performers along with experts and intermediate performers. The proposed ASM technique shows promising alignment and missed/anomalous notification results over this dataset.

5.1 INTRODUCTION

Long-term human actions like warm-up exercise, Sun Salutation, dance performances, etc. consist of time-sequential postures. While performing such actions, the amateur performers tend to miss or wrongly perform a few segments of these long term action sequences. Wu *et al.* [2015a]; Soran *et al.* [2015] proposed techniques to identify missed actions in a long-term sequence. Wu *et al.* [2015a] used a patching-based approach to find missed actions. The sequences are divided into sub-actions using segmentation algorithms followed by sub-actions relation learning from the training dataset i.e. the relative time, when the two sub-actions are executed, is recorded. A test sequence is then divided into constituting sub-actions. The sub-action that is not performed from the set of learned sub-actions is patched at every sub-action transitions. The location at which the likelihood of the missed action occurrence is the highest is reported as the location of the missed action segment. Soran *et al.* [2015] performed the same task of missed action detection using graph-based approach. Using the training videos, the transition weights of the sub-segments are learned. The missed action is reported in case the test performance does not follow the shortest path in the graph to achieve the task.

These baselines perform well while reporting missed actions, however they are strictly applicable to scenarios where there is only a possibility of missed actions. They do not generalise to conditions when there is a possibility of anomalous sub-actions too. In such a case the segmentation algorithm cannot identify anomalies as they can strictly consider modeled actions only. Further,

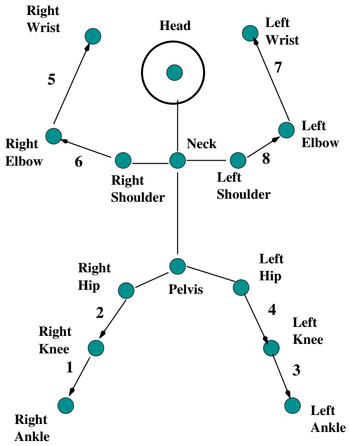


Figure 5.1 : Pose-vector representation

the relative times of even the correctly performed sub-actions are not honoured in such a scenario. Anomalous actions in between a performance can shift all succeeding sub-action start times.

In this chapter, we discuss a technique that can identify both missed and anomalous actions in a long term action sequence. We compare the performance of people to the gold performance instances and provide a feedback to performers on where they miss a posture sub-sequence or perform an anomalous pose sequence using a string matching technique. The proposed framework has been tested on 15 Sun Salutation sequences that contain a mix of such performances. Results are compared with those obtained using the Dynamic Time Warping(DTW) algorithm [Sakoe and Chiba [1978]], a time-series similarity measurement that minimizes the effects of shifting and distortion in time to detect similar performances.

5.2 PROPOSED APPROACH

In this section, we first define how we represent action sequences in terms of pose sequences and then propose our method to find missed and anomalous segments in the video using Approximate String Matching technique.

5.2.1 Pose Estimation

We use the stacked hourglass network [Newell *et al.* [2016]] for human pose estimation. This deep pose model gives state-of-the-art pose estimates over two benchmark datasets, FLIC and MPII Human pose dataset. For each frame the network estimates a pose with 16 joint locations (right ankle, right knee, right hip, left hip, left knee, left ankle, pelvis, neck, thorax, head, right wrist, right elbow, right shoulder, left shoulder, left elbow, left wrist). The joint locations of a pose are normalized relative to the head position thus making them translation invariant [Pirsiavash *et al.* [2014]].

Let $p_x^{(j)}(t)$ be the x component of the jth joint in the tth frame then the normalized joint has its x coordinate given as $s_x^{(j)}(t) = p_x^{(j)}(t) - p_x^h(t)$, where $p_x^h(t)$ is the x coordinate of the head location of the human. Further, the normalized joint points are represented using 8 vectors (Figure 5.1) of the form : $(rK \to rA)$, $(rH \to rK)$, $(lK \to lA)$, $(lH \to lK)$, $(rE \to rW)$, $(rS \to rE)$, $(lE \to lW)$, $(lS \to lE)$, where K,A,H,E,W,S denote knee, ankle, hip, elbow, wrist and shoulder respectively and prefix l, r indicate left or right.

As an example, vector v_1, that connects *right knee* and *right ankle*, is given by

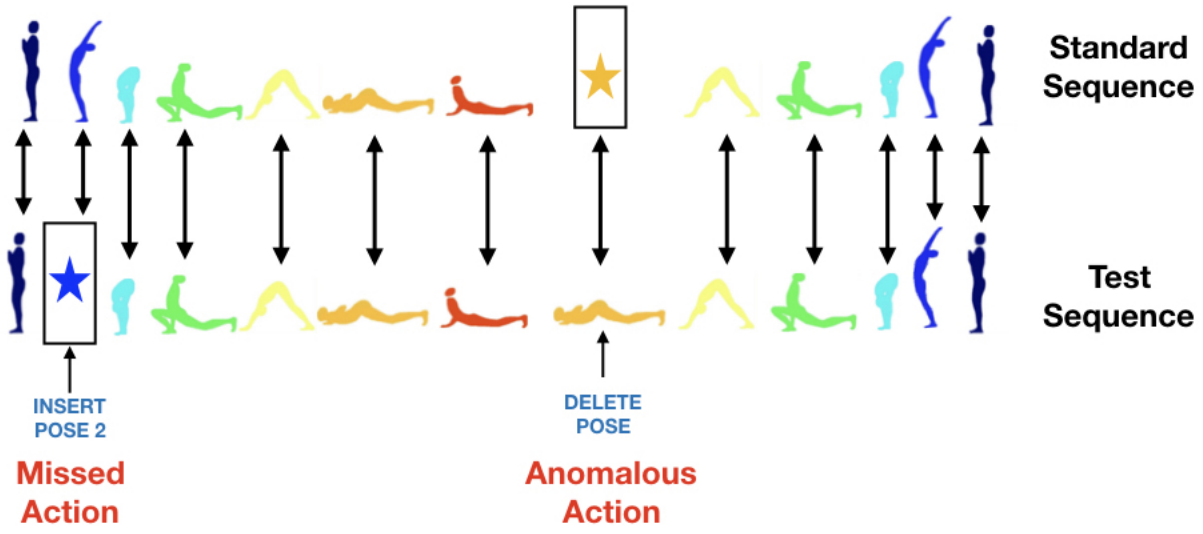


Figure 5.2 : Approximate String Matching illustration

$v_1 = [v_x \ v_y] = [(s_x^{rK} - s_x^{rA}) \ (s_y^{rK} - s_y^{rA})]$. Finally we represent the poses as a vector of 8 angles (RAnkle-RKnee-RHip; LAnkle-LKnee-LHip; LShoulder-LElbow-LWrist; RShoulder-RElbow-RWrist; LElbow-Neck-Head; LKnee-Pelvis-Knee; Neck-Pelvis-RKnee; Neck-Pelvis-LKnee) computed from the dot products of these vectors.

5.2.2 Approximate String Matching algorithm

Approximate String Matching (ASM) has been originally used to solve string matching problem. Given the test video Q composed of pose symbols $q_1, q_2, q_3, ..., q_M$ and the standard template video A with pose symbols $a_1, a_2, a_3, ..., a_N$, ASM finds the smallest number of edit operations that can transform Q into A i.e. ASM finds how test pattern Q is generated from the standard sequence A by calculating the minimum edit distance between Q and A. Let $d_e(i, j)$ denote the minimum edit distance to transform the first j symbols of Q into the first i symbols of A. At each symbol q_j, the editing operations are as follows :

1. *Substitution* : The pose symbol a_i is approximately matched with pose symbol q_j or is substituted by pose symbol q_j with an additional cost $\delta(a_i, q_j)$.

2. *Insertion* : There is an extra pose symbol a_i in A which needs to be inserted into Q with an insertion cost $\delta(\varepsilon, a_i)$.

3. *Deletion* : There is an extra pose symbol q_j in Q which needs to be deleted from Q with a deletion cost $\delta(q_j, \varepsilon)$.

ASM can be solved using dynamic programming and the edit distance at grid (i, j) is defined as :

$$d_e(i,j) = \begin{cases} d_e(i-1,j-1), & \text{if } cost(a_i,q_j) \le th \\ min\begin{pmatrix} d_e(i-1,j-1)+\delta(a_i,q_j) \\ d_e(i,j-1)+\delta(\varepsilon,a_i) \\ d_e(i-1,j)+\delta(q_j,\varepsilon) \end{pmatrix}, & \text{otherwise} \end{cases}$$

i.e. the edit distance at grid (i, j) remains unaltered if the difference between the two poses a_i and q_j is less than a threshold th which is set to 0.005 for our experiments. This difference between the poses $cost(a_i, q_j)$ is the euclidean distance between the 8 dimensional angle vectors. It is important to note here that in case of viewpoint variations this similarity measure would fail and in such cases the similarity can be derived from the field of Epipolar geometry as has been described by Rao *et al.* [2003].

The three operation costs - insert, delete and substitution costs are all set to $1, 1$ and 2. The substitution cost is set higher than the insert and delete operations to avoid sequence transformation in case of missing actions or anomalous actions.

Example. - Lets consider two sequences: an original sequence - 01010 and a test sequence - 1010. If we set a low substitution cost, the test sequence would be transformed into original sequence using 4 substitution operations (replace 1010 with 0101) followed by 1 insertion operation (inserting a 0 at the end). While setting a higher substitution cost will result in a conversion with only 1 insert operation of adding a 0 in the beginning of the test sequence.

Once all the operations needed to transform the test sequence Q to the standard sequence A are determined, the next step is to interpret these operations. A burst of insert operations implies that there is action segment which is missing in the test sequence Q at location s, the starting point of the burst of insertions. Similarly, a burst of delete operations implies that there is action segment which is anomalously performed in the test sequence Q at location s, where the burst of deletions start. This segment does not exist in the standard sequence A and is thus anomalous. Substitution operations occur when the poses in the two sequences are not significantly different i.e. a little adjustment of the pose of the performer is sufficient. Figure 5.2 illustrates the different operations and their interpretations regarding missed actions and anomalous actions on our standard sequence and one of the test sequences from the dataset.

DTW has been used to align similar movements [Su *et al.* [2014]; Hu *et al.* [2015]] to formulate action similarity scores that measure the difference between two time series with different durations. In the next section we examine how conventional DTW algorithm can be used to find missing and anomalous action segments and discuss the problems encountered by this technique.

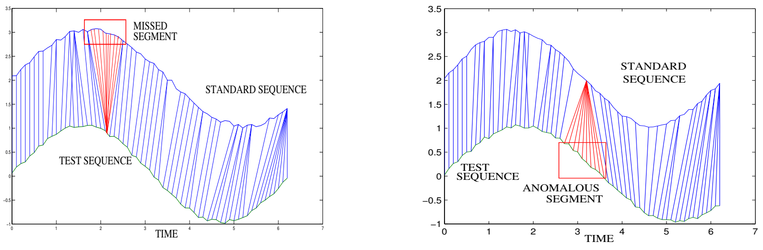
5.3 CONVENTIONAL DYNAMIC TIME WARPING

Given a test action pose sequence Q and a compared action video A, our system aims to find: 1) all video segments in the standard action sequence A that are missed by the performer during his performance Q. 2) all video segments in the performer's action sequence Q that do not occur in the standard sequence A and are anomalous. This requires aligning the two sequences and reporting whenever there are misalignments.

Dynamic Time Warping is a widely used exemplar based sequence matching approach. It is a nonlinear time warping scheme that aims to find the best warping function between any two input signals which gives the minimal total distance. It is tolerant to some degree of time variation between the sequences. The technique uses some constraints to reduce the search space which are -

- **monotonicity constraint** - that prevents the warping path from going back in time axis
- **boundary conditions** - that limits the warping path to start from the first time instance and end at the last time instance for both the test and the standard sequences.

Given a test sequence Q composed of poses $\{q_1, q_2, q_3, ..., q_M\}$ and a compared action video sequence A containing poses $\{a_1, a_2, a_3, ..., a_N\}$, a DTW table of size $M \times N$ is created and the boundaries



(a) (b)

Figure 5.3 : DTW alignment: (a) with missed actions (b) anomalous actions

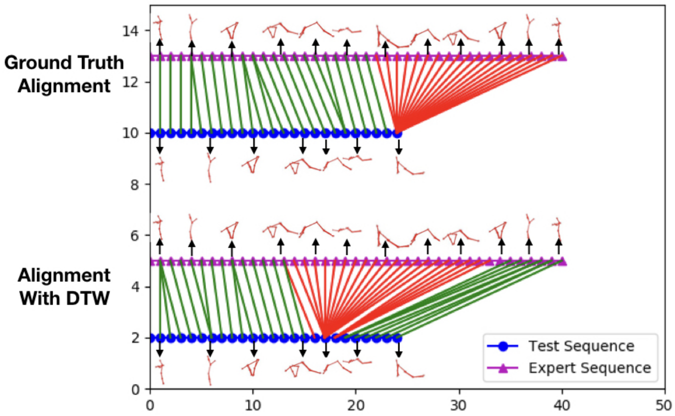


Figure 5.4 : Misalignments due to Dynamic Time Warping

are set as infinity. For $1 \leq i \leq M$ and $1 \leq j \leq N$, each grid (i, j) is filled with a minimum warping distance defined by

$$d_w(i,j) = min\begin{pmatrix} d_w(i-1,j-1) \\ d_w(i,j-1) \\ d_w(i-1,j) \end{pmatrix} + cost(i,j)$$

The DTW method backtracks from the end grid (M, N) to the start grid (1, 1) and construct the entire alignment path which is invariant to temporal transformation.

The existence of missing action segments is marked by the existence of a single frame of the performer's sequence Q aligned to multiple frames from the standard sequence A with count $> th$. Likewise the existence of a single frame of the standard sequence A aligned to multiple frames with count $> th$ from the performer's sequence Q marks the beginning of anomalous action segments as can be seen in Figure 5.3.

However, the alignment of two sequences by the conventional DTW in the presence of such segments is not appropriate. The boundary conditions force the initial and end frames of both

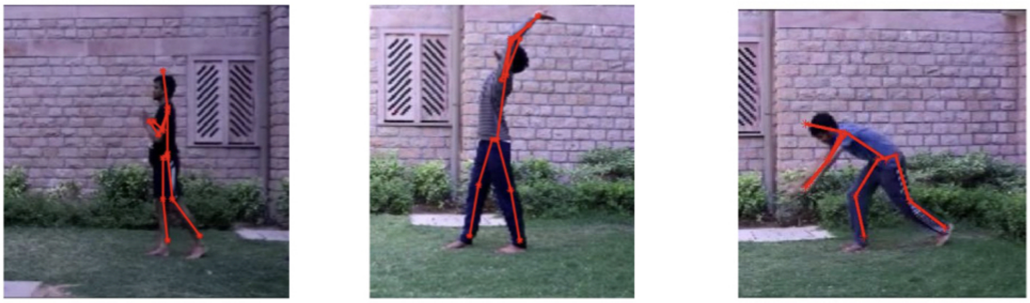


Figure 5.5 : Anomalous Poses in Amateur Videos

the sequences to match to each other leading to misalignment in the rest of the sequence. The misalignment due to boundary conditions can be seen in Figure 5.4 where in a) alignment between two sequences : standard sequence $\{123456754321\}$ and test sequence: $\{1234567\}$ is illustrated. The boundary condition of DTW forces both the sequences to align from the first frame and finish aligning at the last frames of both the sequences. This leads to misaligned action segment 5 and sequence $6, 7, 5$ in the beginning and action segments $6, 7$ of test sequence with $3, 2, 1$ of the template at the end thus resulting in action segments that are incorrectly classified as an anomaly or a missed action (false alarms and missed alarms).

This however is not the case in the ASM technique which is not constrained to begin matching at the initial frame or match the last frames of the two sequences. If there are some action segments missed by a performer at the beginning of the execution, a sequence of insert frame operations are performed until a correctly matching segment is seen. This insert frame operations signify a missed action segment. Similar is the case for anomalous actions, where the delete frame operations are performed until the frames of query sequence and standard gold sequence are matched.

5.4 EXPERIMENTS AND RESULTS

Dataset : We developed a second version of the Sun Salutation with 5 standard templates and 15 test sequences that comprised of sequences with missed action, anomalous segments and few correct ones too. Some of the anomalous frames are as shown in Figure 5.5.

Evaluation : We evaluate the proposed system as a notification module. The accuracy of our notification system can be measured in terms of precision/recall metric. We sub-sampled frames with a difference of 10 frames from the videos for testing. The correctness of a notification is measured in a δ neighborhood of boundaries of groundtruth missed or groundtruth anomalous sub-actions. This means that if a notification of missed segment or anomalously performed action is given in the δ neighborhood of the groundtruth time, it is counted as correct. In our experiments, δ is set as 2 frames, i.e. roughly within 1 second of when it was missed based on the frame rate of 25 fps before sub-sampling.

Tables 5.1 and Table 5.2 list the accuracies of notification module towards missed action and anomalous action detection. It can be seen that the Approximate String Matching technique can correctly notify for all missed action and anomalous segments while the conventional Dynamic Time Warping technique fails to report the same. The lower precision recall in case of DTW is mostly attributed to the misalignment caused due to boundary conditions. Further Approximate String Matching approach has a low precision value for anomaly detection because incorrect posture estimations also leads to false anomalous segment alarms (Figure 5.6).

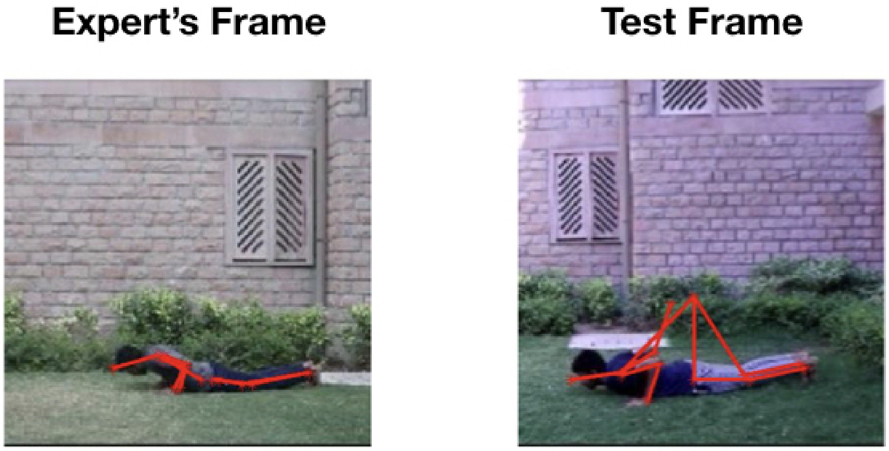


Figure 5.6 : False Alarm for anomalous segments due to incorrect pose estimation

Table 5.1 : Performance accuracy for missed action notification (under tolerance of 25 frames)

Technique	Precision	Recall
ASM	**1**	**1**
DTW	0.71	0.71

Table 5.2 : Performance accuracy for anomalous action notification (under tolerance of 25 frames)

Technique	Precision	Recall
ASM	**0.667**	**1**
DTW	0	0

Though Approximate String Matching gives reasonable performance on missed action and anomalous action detection, it is not the best solution to compare a test sequence with the expert templates. For any action there are many templates equally correct but vary from each other. These variations can result due to speed or posture flexibility. Thus different templates give different edit distance for the same test sequence. For example we considered 5 templates from the same person and compared the test sequences from these templates. The edit distance for the different test sequence were as shown in in Table 5.3. Thus comparison to a single template is not a good solution and we need a solution that adapts an expert to the test performer's speed and then performs matching.

Table 5.3 : Number of Edit Operations of different test sequences with 5 templates of same expert

	Test Sequences														
Expert Template	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	14	17	24	20	24	27	16	15	29	13	20	14	18	19	21
2	12	14	23	23	27	26	14	16	27	14	20	10	15	17	20
3	11	13	21	22	28	27	18	21	33	18	20	13	16	16	20
4	16	1	25	26	28	33	20	20	33	20	19	17	21	21	22
5	15	18	23	25	26	27	17	20	32	18	20	17	20	20	24

5.5 CONCLUSION

In this chapter, we have attempted to make a notification module that can report for missed and anomalous action segments in the performance. We demonstrated how the string matching techniques can be extended to pose sequence matching and detect missed and anomalous action segments in the performances and compare its performance with the baseline Dynamic Time Warping technique for alignment. It is seen that the ASM technique successfully notifies all missed and anomalous actions in the videos while the Dynamic time warping technique fails to align properly and gives incorrect notifications due to its boundary conditions. Further, we saw that a single template cannot be treated as a standard to compare, and thus, instead of a single template matching solution we need a model that learns all expert sequences and then performs a comparison of the test sequence. We provide a solution to this problem in the next chapter where we develop an autoencoder-based model that learns to construct all expert sequences and then infer the skill of a test sequence based on how well it can be reconstructed using the learned model.

…