NrityaGuru: A Dance Tutoring System for *Bharatanatyam* using Kinect

Achyuta Aich, Tanwi Mallick, Himadri B G S Bhuyan, Partha Pratim Das, and Arun Kumar Majumdar
Email: {send2aich, tanwimallick, himadribhuyan}@gmail.com, {ppd,akmj}@cse.iitkgp.ernet.in

Indian Institute of Technology Kharagpur, India, 721302

Abstract. Indian Classical Dance (ICD) is a living heritage of India. Traditionally *Guru*s (teachers) are the custodians of this heritage. They practice and pass on the legacy through their *Shishya*s (disciples), often in undocumented forms. The preservation of the heritage, thus, remains limited in time and scope. Emergence of digital multimedia technology has created the opportunity to preserve heritage by ensuring that it can be accessible over a long period of time. However, there have been only limited attempts to use effective technologies either in the pedagogy of learning dance or in the preservation of heritage of ICD. In this context, the paper presents *NrityaGuru* – a tutoring system for *Bharatanatyam* – a form of ICD. Using Kinect Xbox to capture dance videos in multimodal form, we design a system that can help a learner dancer identify deviations in her dance postures and movements against the prerecorded benchmark performances of the tutor (*Guru*).

1 Introduction

Till date ICD has been passed on to the students by the teacher, from one generation to the next, through the traditional method of *Guru-Shishya Parampara*. We focus on *Bharatanatyam* - a specific form of Indian Classical Dance. To learn *Bharatanatyam* one has to go to a guru, watch her / him perform the steps and then mimic. During the reproduction, the guru provides feedback on the performance of the learner to help her correct the steps. However, while the learner needs to practice these steps at home she neither has the benchmark performance to mimic nor the feedback to correct. Recording the performance of the guru can help getting the benchmark to follow, but instructional feedback remains an open issue.

In this paper we build *NrityaGuru* – an autonomous tutoring system to provide real-time instructional feedback about the correctness of *Bharatanatyam* as performed by a learner. To keep the complexity of the problem manageable, we work only with *Adavu*s of *Bharatanatyam*. An *Adavu* is a basic unit of

Bharatanatyam performance comprising well-defined sets of postures, gestures, movements and their transitions, and is typically used to train the dancers.

The system builds on the skeleton tracking of Kinect as a computational model for *Adavu*s. Using Kinect recording (skeletal as well as RGB videos) of benchmark performances of *Adavu*s by an expert; we employ different approaches to align the time-scales of the learner and the expert to first determine the best match of steps, and then to detect when and how the steps of the learner deviates from the expert. Also, we provide specific feedback on the steps and rate the performance overall.

In Section 2 we discuss related work in tutoring and identify the requirements in Section 3. The system architecture is outlined in Section 5. After discussing the results in Section 6, we conclude in Section 7.

2 Related Work

Over nearly a decade, there have been several attempts, mostly in non-Indian dance forms, to develop autonomous tutoring systems with variety of approaches, sensors, and features. We classify these below.

Virtual / Conceptual Models: In [12], [13] Nakamura et al. proposed a dance movement training systems in 2005 where a user learns to dance by imitating the model (benchmark) dance demonstrated by a virtual teacher. This just supports demonstration. There is no feedback as the user's dance is not captured. Ramadoss et al. [14] made a proposal for a tutoring system to store and retrieve dance data from *Labanotation*.

Force Sensors: Drobny et al. [5], in 2009, developed a system which acquires data from force sensors mounted under the dancers' feet, detects steps, and compares their timing to the timing of beats in the music and help the dancer stay in sync with the music. Force sensors are intrusive and cannot be used for bare-foot dance forms as in ICD.

RGB Videos: In [4], [7], some work was done on storing and synthesizing choreography from RGB video though no full-scale tutoring system was built.

Kinect: Since the introduction of Kinect in 2010, tutoring systems have become more viable and effective. Kinect is the first sensor of its kind that is low-cost, non-intrusive, and multi-modal in audio, RGB-D video and skeleton streams. Understandably there has been proliferation of activities. Efforts include the usage of Kinect based skeleton tracker by Alexiadis et al. [1], Essid et al. [6], and Anderson et al. [2] to develop systems that automatically / semi-automatically evaluate performances of a dancer against a benchmark and provide visual feedback to the performer. Further, Marquardt et al. [11] proposed the Super Mirror that combines the functionality of studio mirrors and prescriptive images to provide the user with instructional feedback in real-time.

All the work reported above focus on western dance forms like Ballet, Samba, or Salsa. There has been no attempt to tutor for any Indian Classical Dance (ICD) form.

3 Requirements Analysis

At the initial stages of development of *NrityaGuru*, we identified a few challenges that defined the requirements.

- *Recorded Data Set:* There is no data set for *Bharatanatyam*. We need to create one (Section 4).
- *Aligning Dance Sequences*: Typically two distinct dance sequences may have different number of frames as they may not be performed for the same duration. Further, the start of skeletal tracking differs across sequences depending on the time taken by the tracking module to detect the dancer and start the tracking process. So we need to align the dance sequences.
- *Identifying Similarity Measure:* Even when two dancers would be performing identical postures, their recorded frames would differ. So we need to define proper similarity measures between two dance frames (learner against expert) based on various parameters.
- *Feedback:* We need way (interface) to provide feedback to the learner in terms of deviations in dance poses. Ideally, this should be real-time. Practically, it could be based on an offline playback. The interface should make it easy to have repeated views and highlight specific areas of deviation.
- *Score Computation:* The correctness of a performance need to be quantified at a frame as well as overall levels in terms of normalized scores.

NrityaGuru system attempts to address the above requirements as explained in Section 5.

4 Data Set Creation and Annotation

No data set for *Bharatanatyam Adavu*s is available for research. Hence, we start by recording 8 different *Adavu*s performed by *Bharatanatyam* dancers – experts as well as learners. A part of the data set is available at [10]. We also get the learners' videos annotated by the experts for deviations.

We use Kinect to capture RGB, depth and skeleton videos at a rate of 30 fps. The skeleton information is used to align videos and compute similarity scores, and RGB and skeleton information are used to provide the visual feedback to the learner. A Kinect skeleton is represented by 20 joint points (Fig. 1) with the root of the Cartesian 3D coordinate system positioned at the hip joint and oriented in alignment with the sensor. Each point is marked as *tracked* or *inferred* – the latter typically denoting the estimate of an occluded body part.

Though the Kinect was found to be mostly adequate for our purpose, yet the following limitations need to be emphasized:

1. Inferred joints of the occluded body parts commonly have substantial noise, which has been taken care of by defining a suitable threshold of error
2. Due to the Limited Field of View dancers needed to perform within a defined space

3. The IR camera is susceptible to noise from various light sources that may contain emissions in the IR band. Hence, special light sources have been used.

Keeping the above in view the following studio setup was created for data recording.

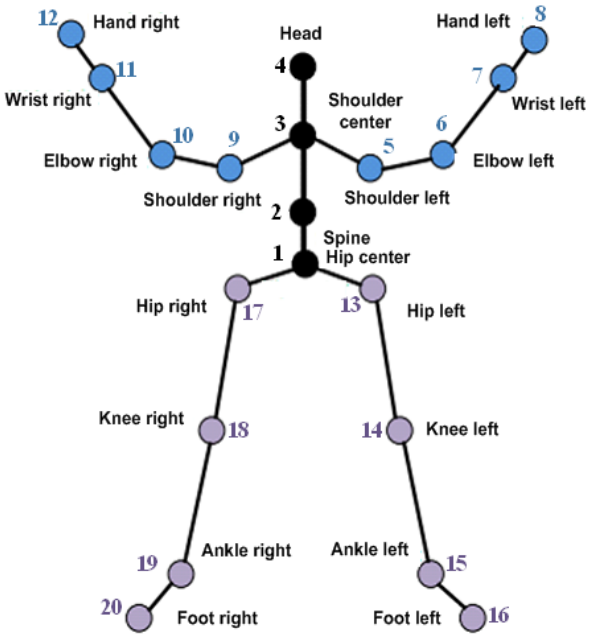


Fig. 1. Different skeletal joints tracked by Kinect

4.1 Studio Setup, Sensors and Tools

The selection of our sensor devices and studio items are listed in Table 1 and our studio setup is illustrated in Figure 2.

The setup is done based on the following considerations:

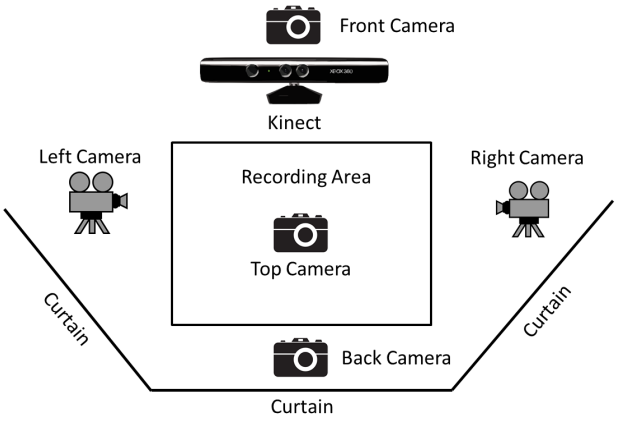
– In spite of the limitations of FoV and DoF, we have decided to use a single Kinect as both the use of multiple Kinects or mirrors inject various kinds of noise and / or artifacts that can be seriously detrimental to the quality required for dance postures and movements.

Table 1. Studio and Sensors as used in recording

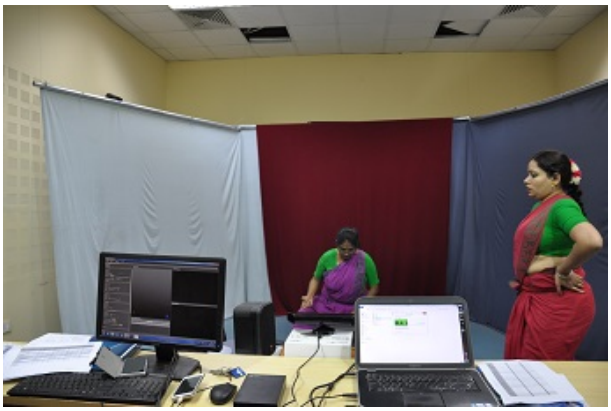
Item	Details
Acoustic Studio	The sound isolation requirement is met by acoustic treatment of the room to keep the reverberation time between 1–1.2 sec. A layer of glass wool having a density of 48 Kg/cubic meter and thickness of 50mm along with 12 mm thick gypsum acoustic panels with NRC 0.85 are used on the walls to achieve the desired isolation criteria. For ceiling, 18 mm thick similar gypsum acoustic boards are used.
Ambient Noise	Studio ambient noise level is maintained within 25 dBA using centralized air-conditioning system
Lights	Philips 4ft 36 Watt Cool white (6500k) Fluorescent lights
Curtains	Grey / Blue / Chocolate colored heavy duty wrinkle free 100% polyester material is used for uniform background and to minimize reflection
RGBD Sensor	Kinect XBox 360 (Kinect 1.0)
Audio Recorder	Zoom H2N Portable Handy Recorder Recording media: SD/SDHC cards Built-in memory: Up to 1 minute in 96kbps MP3 format Mic arrangements: 90° X/Y stereo, MS stereo Microphone types: Directional & Bidirectional (MS side mic) Maximum sound pressure input: 120 dB spl Stereo / 4ch uncompressed PCM, Compressed MP3 Input gain: 0 to +39 dB Rated output level: -10 dBm Headphones: 20 mW + 20 mW Built-in speaker: 400mW, 8O, mono

- The dancer is provided a 3m by 3m area at a distance of roughly 4m from Kinect to perform. This was found to be adequate[1] for performing the *Adavu*s. Also, the dancers rehearse in the space before the recording.
- Unnecessary space in the background is constrained by three backdrop screens at about 45° angle with the imaging plane (Figure 2(a)). This ensures that the maximum depth values do not vary widely, remains within Kinect's DoF, and thus limits the depth noise.
- All shiny / specular surfaces, mirror etc. are avoided in the studio. The backdrop is made with special material suitable for high quality imaging both in RGB and in depth.
- The lighting is done with uniformity to minimize shadows. This ensures good quality for RGB images.
- The audio (a *Sollukattu*) is first recorded with an Audio Recorder. The audio is played back while the dancer performs and Kinect records the audio. This helps orthogonalize the video against the audio and ensures that all *Adavu*s

[1] Such a setup, however, may not work effectively for a performances with a lot of movement.



(a) Schematic View



(b) Front View

(c) Rear View

Fig. 2. Set-up for Recording of *Adavu*s

having the same *Sollukattu* use the same audio file (identical beat structure etc.)
- The Studio is acoustically designed to minimize various audio noise including echoes.

5 System Architecture

The architecture of *NrityaGuru* is shown in Fig. 3. At a time it takes two videos – one pre-recorded by an expert (E) and the other recorded by a learner (L) who is trying to follow the performance of the expert. We attempts to align the videos frame-by-frame so that the learner's performance in a frame can be compared against the corresponding (matching) frame of the expert's performance. If the postures in these matching frames significantly differ, we declare wrong performance by the learner and highlight on the interface (User interface, 5.4). Treating the videos as sequences of frames, we first align the start frames of the sequences by detecting the movement in the first beat. Next we match and align frames pairwise between the two sequences using Dynamic Time Warping (DTW). For matching we define suitable measures. If the measure exceeds a threshold we declare wrong performance. We explain the steps below.

5.1 Alignment of First Frames

To align the first frames between two videos, we note that both the expert and the learner perform to the same musical beats and start the performance with a specific starting movement of stamping a foot. So one option is to analyze the audio of the music, detect the first beat and then align based on the audio. However, in the present work we do not use the audio clue. Rather, we note that at the time of the first beat, the dancer raises her foot (usually right) and puts it down to stamp on the floor. We estimate the velocity of the corresponding joint (ankle right) and detect a zero crossing in its vertical component (Fig. 4) with joint moving up treated as positive. To eliminate errors due to small movements, we use a threshold. Hence in the figure we ignore the crossing around frame 10 and detect the one around frame 100 as the starting. Starting frames so detected in each video ($e_s \in E$ and $l_s \in L$) are used as the first alignment pair.

5.2 Dissimilarity Measure

To match the frames (in DTW and in scoring) we define two similarity measures between every pair of frames from the two videos. The first measure is based on angular dispersion at five major joints while the second is based on the difference in directions of movement of each of the 20 joints. Final measure is computed as a weighted sum of these two measures.

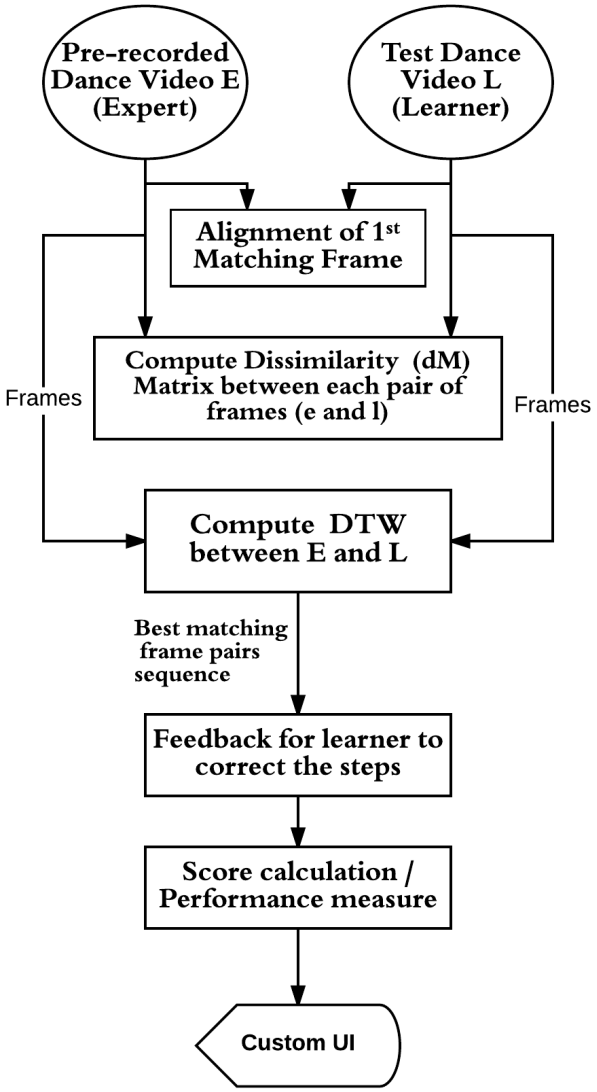


Fig. 3. Architecture of *NrityaGuru*

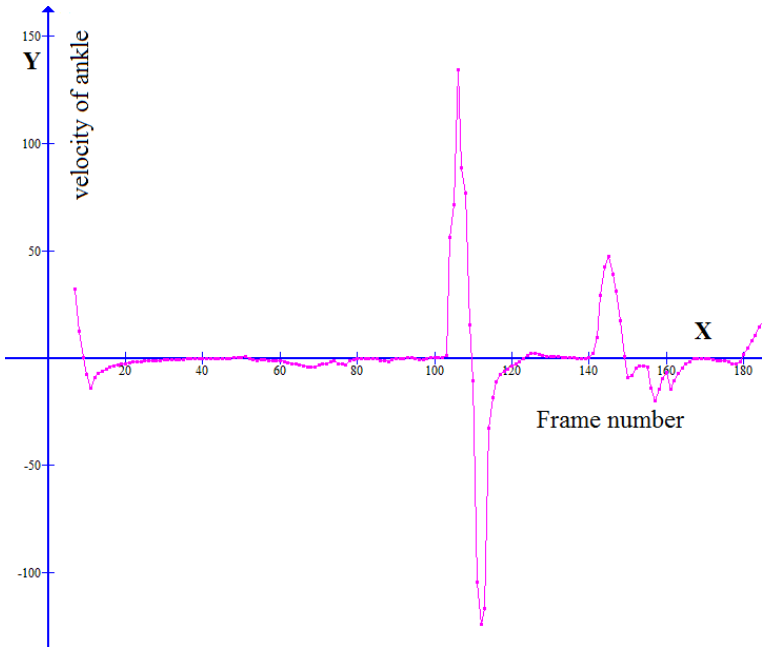


Fig. 4. Vertical component of the velocity of the right ankle joint

Angles at Joints For any skeletal frame we consider angles formed by bones at 5 joints[2]: HKA_L & HKA_R, SEW_L & SEW_R, and LH_HC_RH. These are marked α_0 through α_4. The dissimilarity dM_1 between two frames $e \in E$ and $l \in L$ of the two dancers is then defined as:

$$dM_1(e,l) = \sum_{i=0}^{4} |\alpha_i(e) - \alpha_i(l)| \tag{1}$$

Velocity of Joints For a frame, we compute the unit velocity vector $\|\hat{u}_j\|$ of every joint $j = 1, 2, \cdots, 20$ by considering 5 preceding and 5 following frames. The dissimilarity dM_2 between two frames e and l of the two dancers is then computed as inverse of the cosine similarity as:

$$dM_2(e,l) = \left(\sum_j \frac{\hat{u}_j(e) \cdot \hat{u}_j(l)}{\|\hat{u}_j(e)\|_2 \|\hat{u}_j(l)\|_2} + \delta \right)^{-1} \tag{2}$$

A small positive constant δ is added in the denominator to avoid division by zero. Finally, the dissimilarity dM is computed as a weighted sum of dM_1 and

[2] HKA: hip–knee–ankle, SEW: shoulder–elbow–wrist, LH_HC_RH: left hip–hip center–right hip

dM_2 as:

$$dM(e,l) = w * dM_1(e,l) + (1-w) * dM_2(e,l) \quad (3)$$

where $0 \leq w \leq 1$. This gives the frame–to–frame score.

Defining the Threshold Since two dancers (frame wise) are not exactly same in the temporal / spatial domain, a set of $Thresholds$[3] are required while comparing the similarity between them. Consider a same joint J of two frame e and l, denoted by $J(e)$ and $J(l)$. We calculate the distance, say ds, between these two similar joints. If the following condition (similarity measure) satisfies for each of the joints of e and l then the two frames e and l are considered similar.

$$ds <= (FrameIndex(e) - LastViewedFrameIndex(l)) * Threshold \quad (4)$$

5.3 Frame Correspondence by Dynamic Time Warping

Next we need to correspond every frame of video L with the best matching frame of video L. To arrive at a best matching frame we use *Dynamic Time Warping* (DTW) [3] algorithm to obtain the matching frame pair sequence. For this we first compute the dissimilarity matrix between pair of frames from both videos and then find the path in the matrix that minimizes the sum of dissimilarity measure dM along that path. In other words we need to find the shortest possible path in the matrix from the first matching frame pair to the last matching frame pair. This path will give us the best matching frame pairs.

Note that a frame of one video may match multiple frames of another video, or vice versa. Fig. 5 shows a plot of the matching frames between dancer-1 (expert E) and dancer-2 (learner L). For example, the frame 1200 of dancer-1 matches (marked by a pair of green lines) multiple frames of dancer-2. In that case we select the frame which is at the shortest distance from the one in expert's video.

5.4 User Interface

NrityaGuru supports a custom user interface (Fig. 6) to visualize the dance of the learner (in sync with the dance of the expert). The learner may freeze (pause) at a frame, go forward / backward in steps, or play back continuously to accurately analyze her dance moves. We provide evaluation on the movements in terms of angular difference. We have 5 windows – HKA_L, HKA_R, SEW_L, SEW_R, and LH_HC_RH (between legs) – for this feedback and one window for score display for the learner to know the percentage of accuracy of her / his performance. A set of minimum (-25^0) and maximum ($+25^0$) angular thresholds determines the extent of possible variations between the real–time angles (of learner) and prerecorded angles (of expert) that can still result in a match. When a miss occurs the corresponding window is displayed in red.

[3] Threshold of sudden change = 0.08, Threshold of fast motion = 1.5, Threshold of slow motion = 0.5

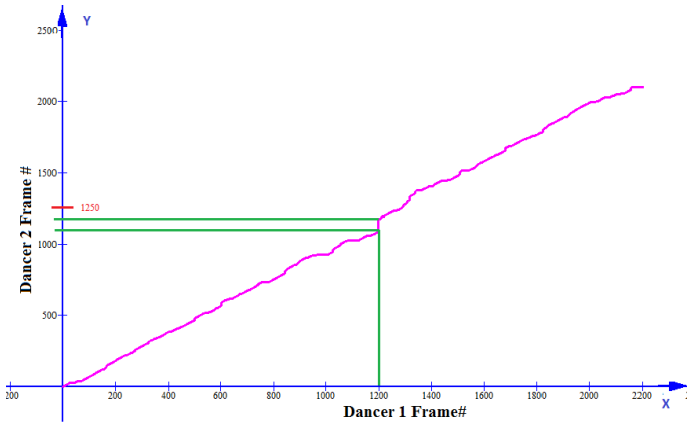


Fig. 5. 2205 frames of Dancer-1 (expert) matched against 2400 frames of Dancer-2 (learner) by DTW

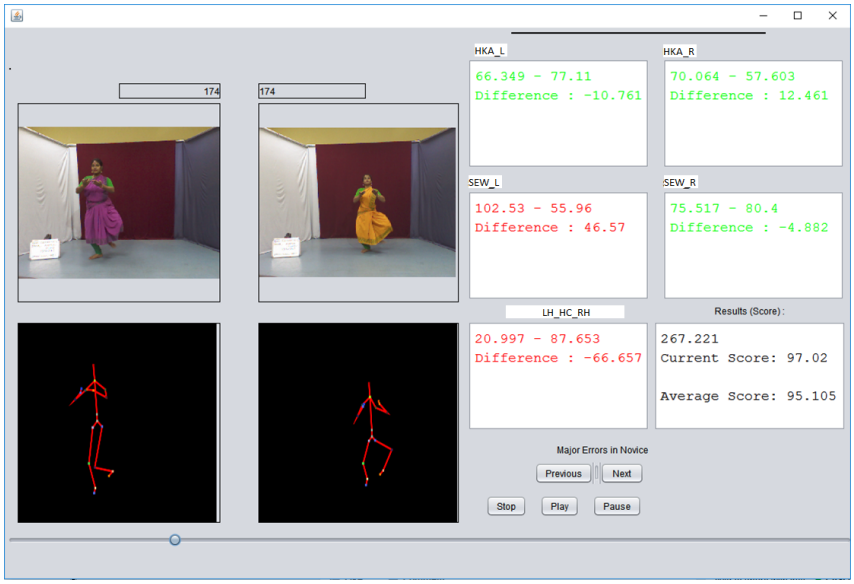


Fig. 6. User Interface of *NrityaGuru*

Fig. 7. Sequence of matching frames between two dancers for a *Pakka Adadu* performance

5.5 Score Computation

The system provides two scores – *Current Frame Score* and *Average Frame Score*[4] to help the learner. For the last matched frame, average frame score is the final score of the performance.

$$Score = Constant/(Constant + dM(e, l)) \tag{5}$$

6 Results

A sample matching for *Pakka Adavu* is shown in Fig. 7.

While it is difficult to provide quantitative accuracy measures for a tutoring system, using the annotated data set (Section 4) we could check the correctness of the flagged (scored red) frames. On the 8 learners' videos, we could achieve 83% accuracy at the frame level. Appropriateness of the performance scores are subjective matter. These have not been independently evaluated by the experts yet.

7 Conclusions

We present *NrityaGuru* – a dance tutoring system for *Bharatanatyam Adavu*s. A learner can use this system to record performances and compare against the correct ones as performed by the experts (prerecorded). The system is low cost, non-intrusive, easy to use, and the first of its kind for Indian Classical Dance.

The implementation of the system has been a non-trivial task. First, the multi-modal data of Kinect has a lot of noise [8] due to various factors. Particularly, the skeletons are often ill-formed (especially when one body part overlaps

[4] Score of match from starting to current frame

on another). Hence, we needed to use various filters (at image as well as skeleton levels) and tune a couple of threshold to stabilize the computations. Second, the Kinect views the dance from one side and all postures (critical for defining correctness) are not completely discernible from the view. Use of multiple Kinect for more complete 3D view has its own issues [9]. Finally, the dancing rules of *Bharatanatyam* are not standardized, allowing for substantial permissible variation at the frame level. In the present system most of these have been craftily handled by tuning the parameters and threshold. However, for scaling up and to generalize to a larger number of *Adavu*s, more in-depth analysis (and partial automated interpretation of the dance form) may be required.

The different types of dance forms have their own set of rules and may require additional features (for example facial features) to be considered. Thus *NrityaGuru* has to be extended to deal with such dance forms. However the angular features of the joints may still be applicable.

The system works in offline mode. We are working to make it real-time. Also, the feedback visualization needs to be improved. We intend to use specific highlight for the errant limbs on the skeleton as well as RGB views.

References

1. Alexiadis, D., Kelly, P., Boubekeur, T., and Moussa, M. B. Evaluating a dancers performance using Kinect-based skeleton tracking. In Proceedings of the 19th ACM international conference on Multimedia (2011), pp. 659–662.
2. Anderson, F., Grossman, T., Matejka, J., & Fitzmaurice, G. (2013, October). YouMove: enhancing movement training with an augmented reality mirror. In Proceedings of the 26th annual ACM symposium on User interface software and technology (pp. 311–320). ACM.
3. Bellman, R., & Kalaba, R. (1959). On adaptive control processes. Automatic Control. IRE Transactions on, 4(2), 1–9.
4. Chan, J. C., Leung, H., Tang, J. K., & Komura, T. (2011). A virtual reality dance training system using motion capture technology. IEEE Transactions on Learning Technologies, 4(2), 187–195.
5. Drobny, D., Weiss, M., & Borchers, J. (2009, April). Saltate!: a sensor-based system to support dance beginners. In CHI'09 Extended Abstracts on Human Factors in Computing Systems (pp. 3943-3948). ACM.
6. Essid, S., Alexiadis, D., Tournemenne, R., Gowing, M., Kelly, P., Monaghan, D., ... & O'Connor, N. E. (2012, March). An advanced virtual dance performance evaluator. In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on (pp. 2269–2272). IEEE.

7. James, S., Fonseca, M. J., & Collomosse, J. (2014, April). Reenact: Sketch based choreographic design from archival dance footage. In Proceedings of International Conference on Multimedia Retrieval (p. 313). ACM.
8. Mallick, T., Das, P. P., and Majumdar, A. K. Characterizations of noise in Kinect depth images. IEEE Sensor Journal 14 (2014), 1731–1740.
9. Mallick, T., Das, P. P., and Majumdar, A. K. Study of interference noise in multi-Kinect set-up. In International Conference on Computer Vision Theory and Applications VISAPP 2014, Proc. 9th National Conference on (2014), pp. 173–178.
10. Mallick, T., Bhuyan, H., Das, P. P., and Majumdar, A. K. Annotated Bharatanatyam Data Set: http://hci.cse.iitkgp.ac.in, 2017.
11. Marquardt, Z., Beira, J., Em, N., Paiva, I., & Kox, S. (2012, May). Super Mirror: a Kinect interface for ballet dancers. In CHI'12 Extended Abstracts on Human Factors in Computing Systems (pp. 1619–1624). ACM.
12. Nakamura, A., Tabata, S., Ueda, T., Kiyofuji, S., & Kuno, Y. (2005, August). Dance training system with active vibro-devices and a mobile image display. In Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on (pp. 3075–3080). IEEE.
13. Nakamura, A., Tabata, S., Ueda, T., Kiyofuji, S., & Kuno, Y. (2005, April). Multimodal presentation method for a dance training system. In CHI'05 extended abstracts on Human factors in computing systems (pp. 1685–1688). ACM.
14. Ramadoss, B., Kannan, R., & Andres, F. (2009). Intelligent Tutoring Systems for Dance Media Environments. In International Advance Computing Conference (IACC 2009).