Dance Composition Using Microsoft Kinect

Reshma Kar[1](✉), Amit Konar[1], and Aruna Chakraborty[2]

[1] Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata, India
rkar317@gmail.co.in, konar.amit@yahoo.co.in

[2] Department of Computer Science and Engineering, St. Thomas' College
of Engineering and Technology, Kolkata, India
aruna.stcet@gmail.com

Abstract. In this work, we propose a novel approach in which a system autonomously composes dance sequences from previously taught dance moves with the help of the well-known differential evolution algorithm. Initially, we generated a large population of dance sequences. The fitness of each of these sequences was determined by calculating the total inter-move transition abruptness of the adjacent dance moves. The transition abruptness was calculated as the difference of corresponding slopes formed by connected body joint co-ordinates. By visually evaluating the dance sequences created, it was observed that the fittest dance sequence had the least abrupt inter-move transitions. Computer simulation undertaken revealed that the developed dance video frames do not have significant inter-move transition abruptness between two successive frames, indicating the efficacy of the proposed approach. Gestural data specific of dance moves is captured using a Microsoft *Kinect* sensor. The algorithm developed by us was used to fuse the dancing styles of various '*Odissi*' dancers dancing to the same *rasa* (theme) and *tala* (beats) and *loy* (rhythm). In future, it may be used to fuse different forms of dance.

Keywords: Dance composition · Computational creativity · Differential evolution · Microsoft *kinect* · Odissi dance

1 Introduction

Gestures are used as alternative form of expression to enrich or augment vocal and facial expressions. As gestures are a part of natural communication, gestural interfaces may be viewed as the pre-requisites of ambient computing [18]. Dance is a special form of gestural expression portrayed in a rhythmic form to generally communicate the context of a music piece [17]. An important part of dance is establishing a smooth flow of dance moves to create a visually appealing sequence of movements.

Dance is specific of different regions, communities and personal styles. Dance schools world-wide teach different dance moves to students, which they are often required to combine to create new dance sequences. It is an important part of dance creation for teachers and self-learning for students. Various techniques have been proposed to ease dance choreography and most of these works have been tested on the

M.L. Gavrilova et al. (Eds.): Trans. on Comput. Sci. XXV, LNCS 9030, pp. 20–34, 2015.
DOI: 10.1007/978-3-662-47074-9_2

dance form 'Ballet' [1, 2]. However, not much work has been done on composing Indian dance forms. In this work, we focus our attention on an Indian classical dance form known as '*Odissi*' which is a classical dance of the state Orissa [10, 11].

Choreographing dance sequences is definitely an art which different people succeed to different degrees. Thus, it would be an interesting task to see how high-end computational algorithms perform at choreographing dance sequences. Naturally, in a visually appealing and easily executable dance sequence, the steps following each other would execute smooth transitions. In our work, we used the same constraint to judge randomly generated dance sequences and finally select the best among them.

At its heart, computational creativity is the study of building software that exhibits behavior that would be deemed creative in humans [3]. Widmer *et al.,* [4] described a computer which can play a musical piece expressively by shaping the musical parameters like tempo and dynamics. In [5] Gervas provided an excellent review of different computational approaches to storytelling. We can find a brief glimpse of the computational attempts at music composition in [6]. Similarly, computational dance composition techniques find mention in the works of [7, 8]. Some works on ballet are discussed in the following paragraphs however their approach is different from ours as they compose dances from pre-defined choreography structures characteristic of 'ballet'. In contrast, dance composition in Indian classical dance forms are more dependent on the context of the music and follow much less rigid sequencing.

A 3-dimensional animation system was developed based on hierarchic human body techniques and its motion data in [9]. Using the proposed system, One can easily compose and simulate classic ballet dance in real-time on the internet. Their goal was to develop and integrate several modules into a system capable of animating realistic virtual ballet stages in a real-time performance. This includes modeling and representing virtual dancers with high realism, as well as simulating choreography and various stage effects. For handling motion data more easily, they developed a new action description code based on "Pas" which is a set of fundamental movements for classic ballet.

An automatic composition system for ballet choreographies was developed by using 3DCG animation in [1]. Their goal was to develop some useful tools in dance education such as creation-support system for ballet teachers and self-study system for students. The algorithm for automatic composition was integrated to create utilitarian choreographies. As a result of an evaluation test, they verified that the created choreographies had a possibility to be used in the actual lessons. This system is valuable for online virtual dance experimentation and exploration by teachers and choreographers involved in creative practices, improvisation, creative movement, or dance composition. The goals, technical novelty and claims of the paper are briefly given below.

Goals:

- To choreograph dance sequence from previously taught dance sequences.
- To study the applicability of a heuristic algorithm in composing dance.
- To be able to automatically identify visually appealing dance sequences.

Technical Novelty:

- A novel measure of fitness of dance sequences is presented.
- Differential evolution is employed to solve the problem of creating a smooth flowing pattern of dance.
- A novel metric of comparison of different forms of dance is introduced.

Claims:

- Heuristic Algorithms can be used to choreograph dance.
- A visually appealing and easy to execute dance sequence can be recognized using the proposed metric for measuring inter gesture transition abruptness.
- Dances of different forms can be quantitatively compared on a common platform using a metric which evaluates the dances forms based on their commonalities.

2 Differential Evolution Algorithm

Proposed by Storn and Prince [12–14] in 1995, Differential Evolution (DE) Algorithm was found to outperform many of its contemporary heuristic algorithms [15]. Following its discovery, the structural simplicity and efficiency of the algorithm attracted researchers, for use in optimization of rough and multi-modal objective functions. Several extensions of the basic DE algorithm are reported in the literature. The De/rand/1 version of the DE algorithm, containing four main steps are outlined below.

(1) Initialization: DE starts with *NP* number of *D*-dimensional parameter vectors, selected in a uniformly random manner from a prescribed search space, given for an engineering optimization problem. The parameter vectors here represent trial solutions of optimization. The *i-th* parameter vector of the population at the current generation *G* is formally represented by

$$\vec{Z}_i(G) = \left[z_{i,1}(G), z_{i,2}(G), z_{i,3}(G), \ldots, z_{i,D}(G)\right] \tag{1}$$

where each component of $\vec{Z}_i(G)$ lies between corresponding components of $\vec{Z}_{\text{min}}$ and $\vec{Z}_{\text{max}}$ given by

$$\vec{Z}_{\text{min}} = \left[z_{\text{min}-1}, z_{\text{min}-2}, z_{\text{min}-3}, \cdots, z_{\text{min}-D}\right]$$

and $\vec{Z}_{\text{max}} = \left[z_{\text{max}-1}, z_{\text{max}-2}, z_{\text{max}-3}, \cdots, z_{\text{max}-D}\right].$

The j-th component of the i-th vector is given by

$$z_{i,j}(0) = z_{j\text{-min}} + rand_{i,j}(0,1) \times (z_{j\text{-max}} - z_{j\text{-min}}), \tag{2}$$

where $rand_{i,j}(0,1)$ is a uniformly distributed random number lying between 0 and 1 and is instantiated independently for each component of the *i-th* parameter vector.

(2) Mutation: For each individual population target vector $\overline{z_i}\,(G)$ belonging to population G, a set of three other randomly selected vectors: $\vec{Z}_j(G), \vec{Z}_k(G)$ and $\vec{Z}_l(G))$ are chosen and arithmetically recombined to create a mutant vector.

$$\vec{V}_i(G) = \vec{Z}_j(G) + F(\vec{Z}_k(G) - \vec{Z}_l(G)) \tag{3}$$

The process is repeated for each parameter vector i in [1, *NP*].

(3) Recombination: Recombination, allows each pair of trial vector $\vec{Z}_i(G)$ to recombine with its corresponding mutant vector $\vec{V}_i(G)$ for i = 1 to NP. Here, recombination is performed position-wise over the length of the parameter vectors. Thus, for trial vectors of length NP, we need to perform NP number of recombination. The principle of recombination for a single position j is given below. Let $rand_{i,j}$ be a uniformly distributed random number in [0, 1] used to select the right element from $V_{i,j}(G)$ and $Z_{i,j}(G)$ into j-th position of the Target vector $u_{i,j}(G)$, where

$$u_{i,j}(G) = \begin{cases} v_{i,j}(G)\, if\ rand_{i,j} \le Cr\, or\, j = j_{rand} \\ z_{i,j}(G)\, otherwise \end{cases} \tag{4}$$

and Cr denotes the crossover rate, which is defined at the beginning of the program once only with a typical value of 0.7.

(4) Selection: In this step, survival of the fittest chromosome is ensured where the offspring replaces the parent vector, if it yields a better value for the objective function.

$$\begin{aligned} \vec{Z}_i(G+1) &= \vec{U}_i(G)\, if\, f(\vec{U}_i(G)) \le f(\vec{Z}_i(G)) \\ &= \vec{Z}_i(G)\, if\, f(\vec{U}_i(G)) > f(\vec{Z}_i(G)) \end{aligned} \tag{5}$$

Repeat from step 2 until stopping criterion, defined on convergence on the best fit member of the trial solutions or convergence of the average fitness over iterations or a fixed large number of iterations are met.

3 Metric to Compare Dance Gestures

Our basic principle in placing dance-gestures adjacently in visually appealing dance choreography is based on a simple comparison metric of these gestures. The dance gestures are video clips of variable duration ranging from 6 to 8 seconds. Since the *Kinect* records at the speed of approximately 30 frames per second, the number of frames captured for each dance-gesture is different. Each video clip, comprising a number of frames, is decoded into a sequence of skeletal diagrams, where each skeletal diagram corresponds to an individual frame (illustrated in Fig. 2). These skeletal diagrams are sketches of the body structure indicated by 3D (three dimensional) straight line segments joining 20 fundamental junctions of the dancer's physique (Fig. 1). Each 3D straight line segment is projected on to XY, YZ and ZX planes as illustrated in Fig. 3 a, b, c, and the slopes of the projected 2D straight lines in the three planes with respect to X-, Y- and Z- axes respectively are computed. These slopes are used as metrics to compare two different 3D straight lines representing the orientation of one given body part (say, the right forearm) present in two frames of two videos. While constructing a new dance video from the existing video clips of shorter durations, we need to match the last frame and the first frame of two consecutive videos, which might come in order to ultimately offer a new dance video. Naturally, the matching of 3D straight lines is required between the last frame and the first frame of each pair of videos to test their possible just-apposition in the final video.

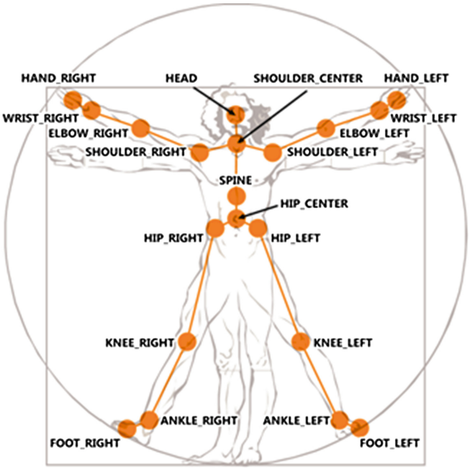

Fig. 1. 20 body joint co-ordinates obtained from Kinect sensor for each frame

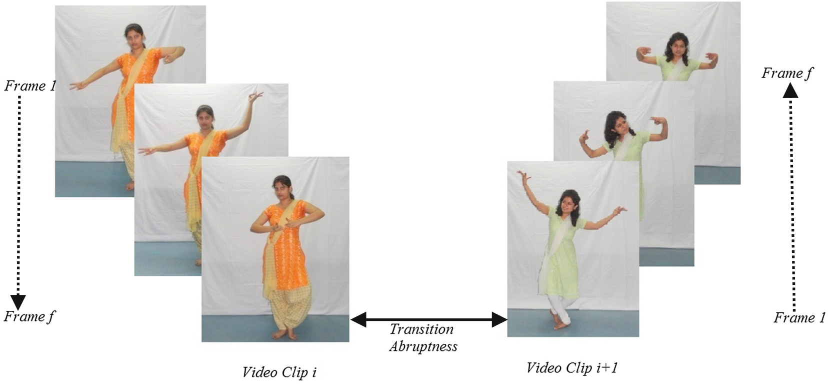

Fig. 2. Video clips of dance gestures illustrating frames. It can be seen that Transition Abruptness is calculated by comparing the last and first frames of consecutive dance gestures i and i + 1 respectively.

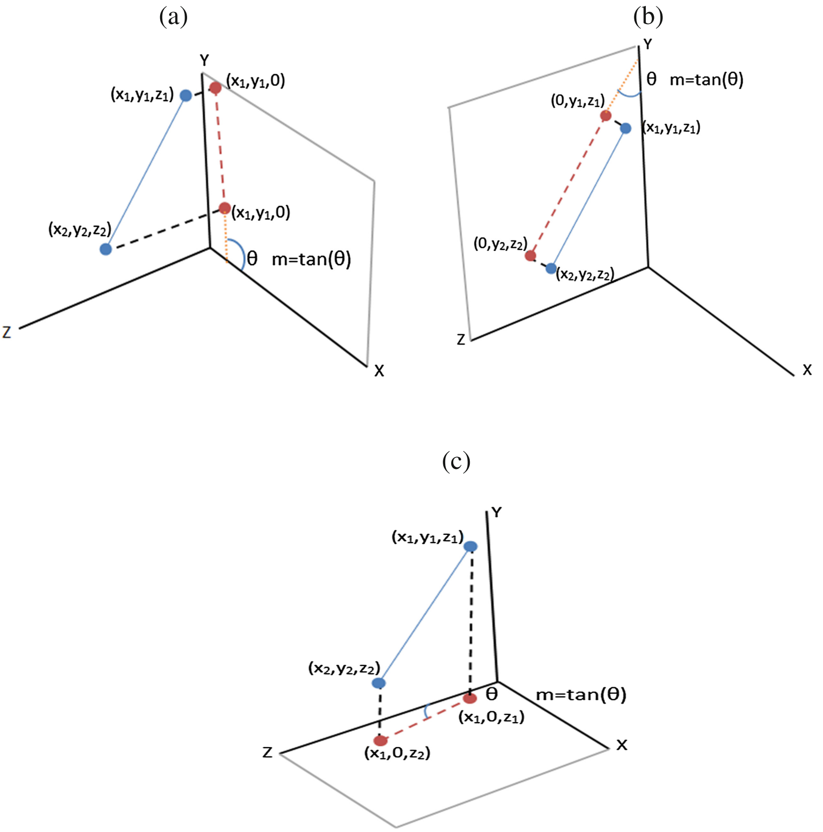

Fig. 3. Calculation of slope of projections of a line on XY, YZ and ZX planes wrt. X, Y and Z axes respectively

A. Definition: We introduced a new term called, Transition Abruptness (TA), to measure the abruptness of inter-gesture transitions in a dance sequence. It is a measure of the total difference between two 3-dimensional body skeletal structures, induced by comparing the angular difference of the projected straight lines of each 3D straight line link of two skeleton structures. TA between two dynamic gestures, G_i and G_{i+1}, each comprising a number of sequential frames, is measured by (6), where $p_{x,i,j,k}$ denotes the slope of the projected straight line k on the X-Y plane with respect to x-axis of the j-th frame of the i-th gesture. The parameters: $q_{y,i,j,k}$ and $r_{z,i,j,k}$) denote slopes of the projected straight line k on the Y-Z plane (Z-X plane) with respect to y-axis (z-axis) of the j-th frame of the i-th gesture.

$$TA(G_i, G_{i+1}) = \sum_{\forall k} \sum_{\forall i,\forall j} \left(|p_{x,i,1,k} - p_{x,i+1,f,k}| + |q_{y,i,1,k} - q_{y,i+1,f,k}| + |r_{z,i,1,k} - r_{z,i+1,f,k}|\right) \tag{6}$$

The *total transition abruptness* of the gesture permutation $G_1, G_2, G_3 \ldots G_n$ is calculated in (7) by summing TA of each two consecutive dance gestures

$$\mathit{Total\,Transition\,Abruptness} = \sum_{\forall i} TA\left(G_i, G_{i+1}\right) \tag{7}$$

4 Selection of Dance Sequences

Evolutionary Algorithms are generally employed to optimize complex non-linear, multi-dimensional objective functions which usually contain multiple local optima. In our experiments, the search surface considered is characteristic of dance-move permutations. The fitness function which ensures the survival of best dance sequences is obtained based on the total abruptness in inter-gesture transitions. In this section, we present a novel scheme of dance composition by selection of dance permutations. The basic structure of parameter vectors used here is of 3K dimension, where K denotes the maximum value of k. The integer 3 appears due to inclusion of three elements: $p_{x,\, i,\, j,k}$, $q_{y,\, i,\, j,k}$ and $r_{z,\, i,\, j,k}$ each K-times in the parameter vector. It is important to note that we have two sets of parameter vectors, the first set for representation of the first frame of each video clip describing a dynamic gesture, and the last set to represent the last frame of the same dynamic gesture. In the evolutionary algorithm we match two dynamic gestures by measuring the total transition abruptness of the last frame and the first frame of any two video clips. The pseudo code of the proposed gesture selection algorithm is outlined next.

```
Pseudo Code
Input: A set of body joint co-ordinates of $n$ dance gestures$G_1, G_2 \ldots. G_5$.
Output: A visually appealing permutation of $m$ dance gestures.
Begin
    1. Randomly initialize a population of NP solutions (dance sequences)
       $\vec{Z}_j(g) = \{G_1, G_2 \ldots., G_5\}$ . Here j runs from 1 to NP and g signifies generation
    2. For Iter=1 to MaxIter
       Begin
           a) Create mutant vector using mutation scheme section 2.
           b) Crossover target vector and mutant vector as outlined in section 2.
           c) Evaluate Fitness by equation 7 i.e.
              $fitness = \sum_{i=1}^{n} Transition\ Abryptness(\vec{G}_i, \vec{G}_{i+1}) \quad \forall G_i \in \vec{Z}$
           d) If trial vector is better (less abrupt) than target
              Then replace trial by target in the next generation
              End if
       End For
    3. Return $\vec{Z}$ after minima value of fitness function converges.
End
```

```
Procedure: Transition Abruptness ( $\vec{G}_i,\vec{G}_{i+1}$ )
    Input: $\vec{G}_i,\vec{G}_i+1$
    Output: Inter-gesture Transition Abruptness TA
    Begin
    Let K be the number of lines in each skeletal diagram
     1. For k=1 to K
          a) Extract $l^k_{G_i}, l^k_{G_i+1}$ where $l^k_{G_i} \in last\ frame\ of(\vec{G}_i)$
             and
             $l^k_{G_i+1} \in first\ frame\ of(\vec{G}_i+1)$

             Here each of the above lines are represented by ,the projections of their
             slopes ($p_{x,i,j,k}$, $q_{y,i,j,k}$, $r_{z,i,j,k}$ discussed preceding eq. 6)

          b) Compute the fitness of each consecutive pair of gestures by Eq.6
             $A_l = TA(G_i, G_{i+1})$

    End For
     2. TA=0

     For l=1 to N-1
          a) $TA = TA + A_l$

    End For
     3. Return TA
```

5 Comparison Metric of Different Dance Forms

The comparison of dance composition algorithms is a difficult task as there is no numerical formula to distinguish the quality of one dance from another. To the best of our knowledge, ours is the first work on *Odissi* dance composition which makes the comparison of our algorithm further difficult. Thus, we also propose a metric for comparison of dance of different forms and use it to compare dance composed by our algorithm with dance composed by other algorithms. Apparently, a good dance composition technique maintains a smooth flow of transitions between gestures and also a high amount of dynamism. The smoothness of inter-gesture transitions can be measured by the calculating *Transition Abruptness* between two consecutive gestures. A smaller *Transition Abruptness* between consecutive gestures indicates a smooth flow of dance. Similarly, if non-adjacent gestures have higher value of *Transition Abruptness* the dance can be said to have more dynamic steps. Thus, we define the *Visual Appeal* of a dance sequence *S* as follows composed of *n* gestures G_i, where *i* ranges from 1 to *n*.

$$Visual\ \ Appeal(S) = \frac{\sum_{i=1}^{n} Transition\ \ Abruptness\ (G_i, G_{i+2})}{\sum_{i=1}^{n} Transition\ \ Abruptness\ (G_i, G_{i+1})} \tag{8}$$

6 Experiments

The Microsoft *Kinect* captures skeletal co-ordinates at an approximate rate of 30 frames per second, with the help of an RGB camera, an infrared projector a monochrome CMOS (complimentary metal-oxide semiconductor) sensor. This configuration allows the *Kinect* to efficiently capture the 3-dimensional skeletal co-ordinates in closed room settings. However, certain precautions are required to be maintained to ensure noise-free collection of data. A white background is maintained to eliminate possible chances of interference of background objects in data collection. To ensure good quality of images, two standing lights are placed facing the dance platform. The dancers are also instructed to avoid wearing clothes which are too loose around the body joints. The subjects should be within an appropriate range of the *Kinect* which is approximately 1.2 to 3.5 meter or 3.9 to 11 ft.

In our experiments, 10 experienced *Odissi* dancers are asked to perform 5 of their preferred dance gestures which are each captured separately with the help of the Microsoft *Kinect*. Thus, we have a total of 50 video clips of dynamic dance gestures. A sequential permutation of 5 of these dance gestures are then selected from a large amount of such randomly created permutations with the help of *differential evolution* algorithm. It is implicit that dancing style of each dancer is dependent on various factors including personal choice and interpretation of dance steps. This lends a unique touch to each of the dancer's gestures though all of them belong to the same dance form. The aim of our experiments is to combine the dance styles of different teachers which has the potential to give rise to interesting patterns and also has future applications in fusing various dance forms. Stills from dance gestures of the different teachers are shown in Fig. 4.

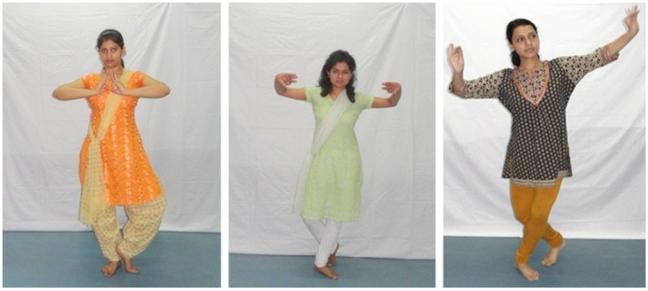

Fig. 4. Gesture stills of different "*Odissi*" dancers

The data captured using *Kinect* represents the bodily movements over time with an approximate distance of 10 cm from ground truth. However, some important considerations must be made before performing analysis of the skeletal co-ordinates obtained.

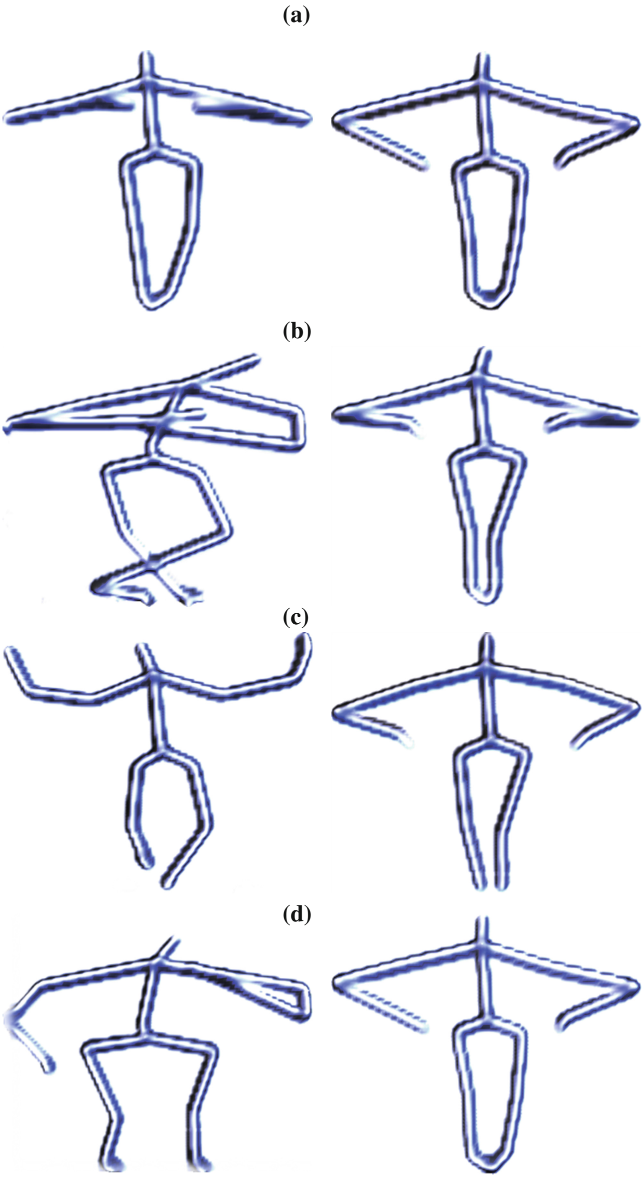

Fig. 5. Inter-Gesture Transition poses

For example two subjects (dancers) may be portraying the same gesture and yet have different X, Y and Z co-ordinates. The main reasons of this are difference in skeletal structure of subjects and variable distance of subjects from *Kinect* camera. To overcome these differences, we have used the slope of the lines in skeletal diagram as the comparison parameter.

In the next phase, differential evolution algorithm is used to evaluate different permutations of the dance sequences and find the best among them. The comparison of dance sequences is based on *total transition abruptness*. The total transition abruptness is simply the sum of all *inter-gesture transition abruptness* $TA = \left(G_i, G_{j+1}\right)$. The comparison for placing the gestures adjacently is based on the first frame of a gesture and the last frame of the preceding gesture. In Fig. 5, we have illustrated the diagrams such four pairs; in Table 1 we described how the *inter-gesture transition abruptness* (total slope differences of the projections of the indicated skeletal diagrams) are able to detect inter move abruptness of the pairs in Fig. 5. In Fig. 7 we have shown a part of one of the fittest generated dance sequences.

Table 1. Analysis of Inter-gesture Transition Abruptness values obtained

Fig. No.	Transition Abruptness (TA)	Analysis
Figure 5a	292.0031	It can be easily seen that the poses are quite close
Figure 5b	350.9825	TA increases as gradual slope changes occur
Figure 5c	955.3705	Drastic Changes in both upper and lower body increase TA
Figure 5d	900.0376	Comparatively more changes in lower body than upper body cause slight decrease in TA

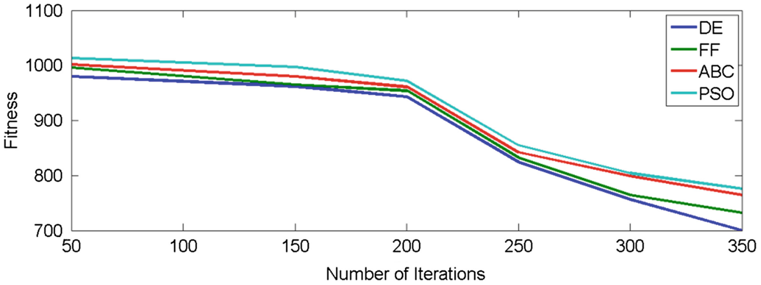

Fig. 6. Value of Fitness Function of different Heuristic Algorithms for Dance Composition, *DE*: *Differential Evolution*, *FF*: *Firefly Algorithm*, *ABC*: *Adaptive Bee Colony optimization*, *PSO*: *Particle Swarm Optimization*.

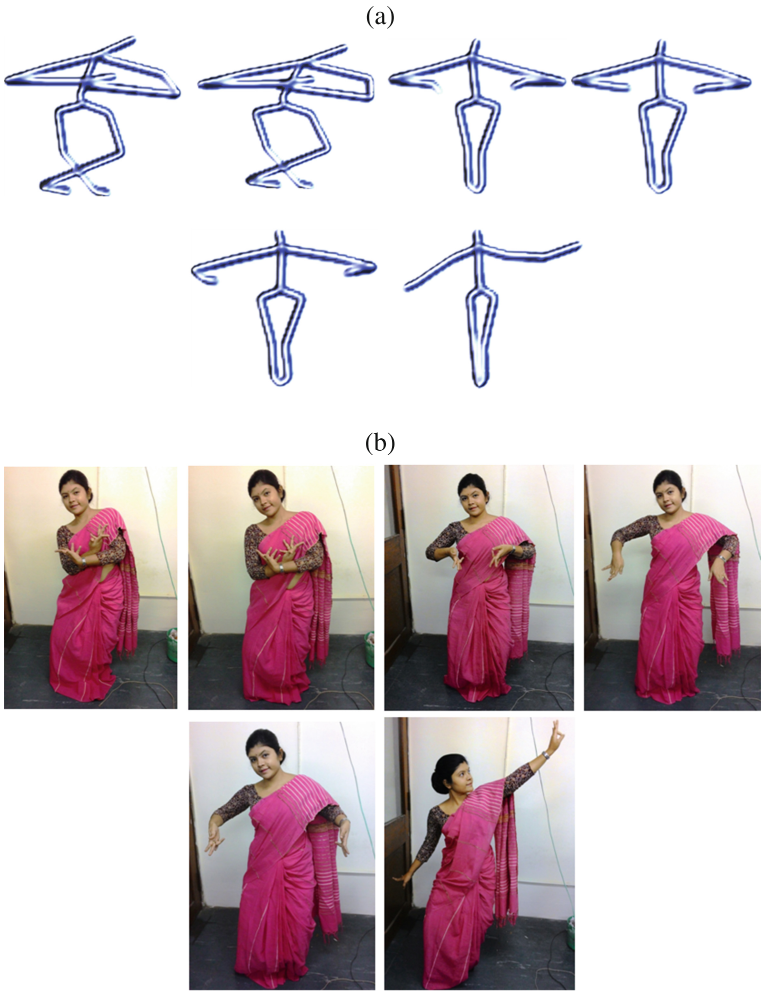

Fig. 7. Subsequence of the final dance sequence generated by differential evolution demonstrated by (a) Skeletal diagrams (b) A dance student's performance

7 Performance Evaluation

Finally, we asked 10 *Odissi* dancers to rate the dance permutation generated by our system on a scale of 1–4 as follows, 1: Bad, 2: Fair, 3: Good, 4: Excellent. The results obtained are given in Fig. 8. The convergence of different optimization algorithms on the dance composition problem is illustrated in Fig. 6, the corresponding data is also given in Table 2. It is seen that *Differential Evolution* algorithm performs better than 3 other algorithms namely *Adaptive Bee Colony algorithm*, *Firefly algorithm* and *Particle Swarm Optimization* for the dance composition problem. Using our proposed metric *visual appeal* outlined in Eq. 8, we compared our dance composition techniques with two other dance composition techniques for different dance forms [1, 16]. Our algorithm performs substantially better than other algorithms on an average as indicated in Table 3.

Table 2. Average fitness value of different heuristic algorithms over different iterations

Iterations	Fitness			
	DE	FF	ABC	PSO
50	980	996	1002	1013
150	962	965	980	997
200	943	954	961	972
250	824	832	842	855
300	756	764	799	804
350	700	732	765	776

Table 3. The comparison of different dance composition algorithms on the basis of *visual appeal.*

Dance composition technique	Average *Visual Appeal* for 3 runs
Ballet Composition [16]	18.67
Contemporary Dance Composition [1]	19.99
Odissi Composition by Differential Evolution	23.44

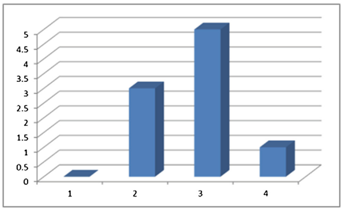

Fig. 8. Performance Evaluation of the Dance composition System as per the opinion of 10 dancers; here 1: Bad 2: Fair 3: Good 4: Excellent

8 Discussion

This paper introduced a meta-heuristic approach to compose dance by optimally selecting inter-gestures movement patterns from multiple dance gestures. The fitness estimate used for dance gesture composition ensures a smooth transition of frames in the dance video. One metric representing visual appeal has been introduced to compare the performance of the proposed Differential Evolution (DE) based meta-heuristic algorithm with existing algorithms of dance composition. The proposed DE-based realization is also compared with other standard realizations including, firefly, PSO and ABC with respect to convergence time, and the results indicate that the DE realization outperforms its partners in the present context.

9 Conclusion

This paper provides a fresh perspective on the technique of dance composition. Dance composers are commonly required to ensure a smooth flow of inter-gesture dance transitions. Our algorithm uses this common know-how and integrates it into the differential evolution algorithm to compose interesting dance patterns. Since the differential evolution algorithm uses a random initial population each time; the chances of dance patterns generated being novel each time is high. The algorithm can be improved by including other constraints in the fitness function which increase the dynamicity of the dance patterns. There is also a wide scope in creating dance composition with context to music-mood as well as lyrics. Apart from this one may also analyse the relative performance of different evolutionary algorithms in dance composition. In a nutshell, our work area relatively unexplored and offers a wide future scope.

References

1. Soga, A., Umino, B., Hirayama, M.: Automatic composition for contemporary dance using 3D motion clips: experiment on dance training and system evaluation. In: International Conference on CyberWorlds (CW 2009), pp. 171–176. IEEE (2009)
2. Dancs, J., Sivalingam, R., Somasundaram, G., Morellas, V., Papanikolopoulos, N.: Recognition of ballet micro-movements for use in choreography. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1162–1167. IEEE (2013)
3. Colton, S., López de Mantaras, R., Stock, O.: Computational creativity: coming of age. AI Mag. **30**(3), 11–14 (2009)
4. Widmer, G., Flossmann, S., Grachten, M.: YQX plays Chopin. AI Mag. **30**(3), 35–48 (2009)
5. Gervás, P.: Computational approaches to storytelling and creativity. AI Mag. **30**(3), 49–62 (2009)
6. Edwards, M.: Algorithmic composition: computational thinking in music. Commun. ACM **54**(7), 58–67 (2011)
7. de Sousa Junior, S.F., Campos, M.F.M.: Shall we dance? A music-driven approach for mobile robots choreography. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1974–1979. IEEE (2011)
8. Jadhav, S., Joshi, M., Pawar, J.: Art to SMart: an evolutionary computational model for BharataNatyam choreography. In: HIS, pp. 384–389 (2012)
9. Soga, A., Endo, M., Yasuda, T.: Motion description and composing system for classic ballet animation on the web. In: 10th IEEE International Workshop on Robot and Human Interactive Communication, pp. 134–139. IEEE (2001)

10. Saha, S., Ghosh, S., Konar, A., Janarthanan, R.: Identification of Odissi dance video using Kinect sensor. In: International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1837–1842. IEEE (2013)
11. Saha, S., Ghosh, S., Konar, A., Nagar, A.K.: Gesture recognition from indian classical dance using Kinect sensor. In: Fifth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), pp. 3–8. IEEE (2013)
12. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
13. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. ICSI, Berkeley (1995)
14. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Springer, Heidelberg (2006)
15. Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Congress on Evolutionary Computation (CEC), vol. 2, pp. 1980–1987. IEEE (2004)
16. Soga, A., Umino, B. and Longstaff, J.S.: Automatic composition of ballet sequences using a 3D motion archive. In: 1st South-Eastern European Digitization Initiative Conference (2005)
17. Patra, B.G., Das, D., Bandyopadhyay, S.: Unsupervised approach to Hindi music mood classification. In: Prasath, R., Kathirvalavakumar, T. (eds.) MIKE 2013. LNCS, vol. 8284, pp. 62–69. Springer, Heidelberg (2013)
18. Kar, R., Chakraborty, A., Konar, A., Janarthanan, R.: Emotion recognition system by gesture analysis using fuzzy sets. In: Swarm, Evolutionary, and Memetic Computing, pp. 354–363. Springer International Publishing (2013)

Springer

http://www.springer.com/978-3-662-47073-2

Transactions on Computational Science XXV
Gavrilova, M.L.; Tan, C.J.K.; Saeed, K.; Chaki, N.; Shaikh,
S.H. (Eds.)
2015, XIV, 201 p. 94 illus., Softcover
ISBN: 978-3-662-47073-2