Automated Analysis and Interpretation of Bharatanatyam Dance

Himadri B.G.S Bhuyan

Under the Supervision of
Prof. Partha Pratim Das

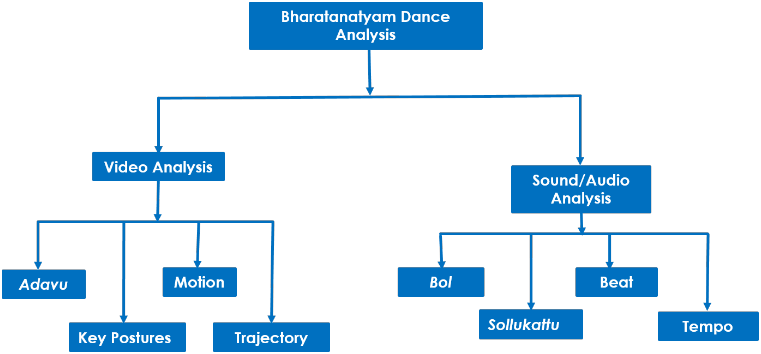
Rajendra Mishra School of Engineering Entrepreneurship
Indian Institute of Technology Kharagpur

Extension Seminar July-2020

May 11, 2020

Outline

Introduction: Analysis of Bharatanatyam



Terminology Associated with *Bharatanatyam* Dance

- **Adavu:** Basic unit of Bharatanatyam
- **Key Postures:** Momentarily stationary well-defined postures occurs within the *Adavu*
- **Key Frames**: Frames associated with a key posture
- **Motion Frames**: Frames associated with motion
- **Bol:** Utterance. A bol is a mnemonic syllable.
- **Sollukattu:** Accompanying Sound Track of an Adavu
- **Beat:** Basic unit of time in music
- **Tempo:** Pace or speed at which a section of music is played

Motivation

- Heritage Preservation
  - Preserving the Knowledge and Practices of Experts (*Guru*s) digitally
- Tutoring System
  - Assist the learner in the absence of the teacher
- Dance Interpretation
  - It inclined towards cognitive domain.
- Dance Synthesis
  - Innovation or a new way of expression: Dance choreography, Creating animated Avatar

Objectives

1. Analysis of Structured Audio of *Bharatanatyam*
2. Analysis of Structured Video of *Bharatanatyam*
3. Study of Synchronization between Audio and Video of *Bharatanatyam*, between different components of *Bharatanatyam*
4. Building the knowledge graph of *Bharatanatyam*
5. Demonstration through sample applications

Work Status

Task	**Prev. Completed Work**	**Recent work / Future Scope**
Data Capture, Extraction and Annotation	• Video:KP and Motion Annotation • Audio Annotation Done	• Motion Annotation, Session-2 to be completed
Bol Detection	• GMM (Gaussian#:8, 15, 30, 50), co-variance type: Diagonal, Spherical, Full, Tied	• Beater independent *Bol* detection
Sollukattu recognition	• Naive Bayes • Linear SVM • Multinomial & Bernoulli Naive Bayes	• Beater independent Recognition
Key Posture Recognition	•Feature: Angle & HOG •Recognizer: SVM and GMM	•To Explore Bayesian techniques
Adavu Recognition Using KP	•HMM •SVM & Edit distance (ED) on Angle Feature • SVM & ED on HOG Feature	• *Adavu* Recognition by selecting random Frames •*Adavu* Recog. Including Motion aspect
Applications	—*NrityaGuru* [2] —Human Postures to *Labanotation* [3]	– •To try on Dance • Automatic Annotation tool
Motion & key Frame Detection	•Non-adaptive & Rule based •Using velocity of skeleton Joints • Adaptive & rule based • ML Approach	**Completed** [1]
Motion Classification	Using velocities of Limb joints	• Using Optical flow as feature in KNN • Using Trajectory of Limb joints • Using HoG/HOOF on RGB data

Table: Work Status

Recent Work

- *Adavu* recognition
  - Feature: Histogram of Optical flow (HOOF)
  - Classifier: SVM
- Motion classification
  - Feature: Optical Flow
  - Classifier: KNN

Adavu Recognition

- *Adavu*: Basic unit of *Bharatanatyam* Dance
- It comprises of *Key postures* followed by some motion
- Approach:
  - **Input:** The Adavu video, the set of RGB frames
  - **Output:** Labelling the type of *Adavu*

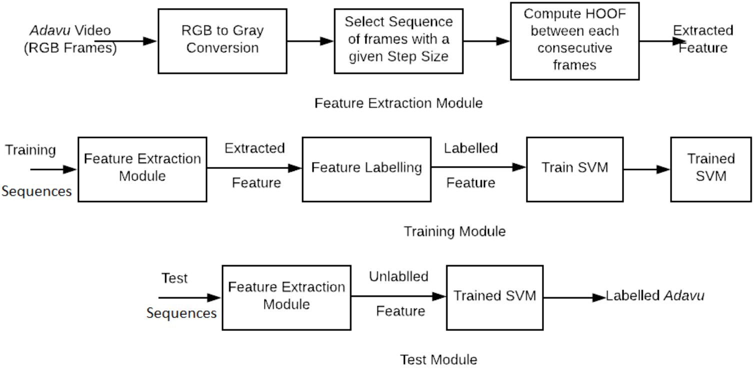


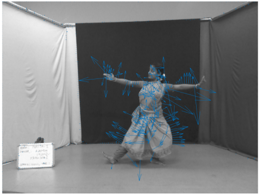
Figure: Work flow

Optical flow

- Input: Two Consecutive Images
- Output: Optical flow/velocity vector (Vx, Vy)=(u,v) of each pixel
- Assumption: Brightness/Intensity doesn't change with time. So I(x,y,t) = I(x+u,y+v,t)
  $\Rightarrow I_x.u + I_y.v + I_t \approx 0$
  We can compute (u,v) using this equation.
  Where I_x and I_y = Image Derivative along X and Y-axis respectively,
  I_t = Difference over 't' or Image Difference.


Previous Image


Next Image


Showing Flow

Data set and Technical details

(b) *Used Data Set*

Adavu	#of Frames D1	D2	D3	#of Adavu Seq. D1	D2	D3
Mettu-1	612	630	595	11+4	26+8	21+6
Mettu-2	555	629	600	20+7	21+6	9+3
Mettu-3	620	640	603	18+5	23+7	11+3
Mettu-4	1065	1069	1035	15+5	15+5	21+7

(b) *Sample Annotation*

KP ID	**Start Frame**	**End Frame**
M1D1B1P02	69	86
M1D1B1P03	104	123
....	....	....
M1D1B16P01	592	612

- Each RGB frame is of size: 480x640
- RGB to Gray Conversion: $0.3 * R(i,j) + 0.59 * G(i,j) + 0.11 * B(i,j)$
- #of frames per sequence (n): 24
- Step Size $= (LastFrame - Annotation(2,3))/(n-1)$
- Optical flow Technique: Lucas Kanade, Feature set: Vx = Vy = 480x640
- Histogram technique: Weighted Binning, #of Bins = 10 (0° to 180°)
- Train Sequence: 75%, Test Sequence: 25%
- **Challenges:**
  - Covering most the frames from start to end in each sequence
  - Each sequence must contain most of the KP and motion frames alternatively.

Result and Analysis

Adavu	Test Accuracy(%) & Miss Matched Adavus					
	D1		D2		D3	
	Acc	Error	Acc	Error	Acc	Error
Mettu-1 (M1)	2/4=50	2M2	5/8=62.5	2M2,M3	4/6=66.66	2M3
Mettu-2 (M2)	4/7=57.14	M1,2M3	4/6=66.66	2M1	2/3=66.66	M3
Mettu-3 (M3)	3/5=60	M2	4/7=57.71	2M2,M1	2/3=66.66	M2
Mettu-4 (M4)	5/5=100	–	4/5=80	M1	7/7=100	–

Table: Accuracy Result & Miss Matched Adavus

Analysis

- Only Mettu-4 shows good accuracy
- The Highest accuracy of Mettu-4 because
  - It has more number of frames, which leads to Bigger step size, the more KP frames get selected as compare to motion frames
  - Presence of unique KPs and motions
- Non of the *Mettu* miss classified with *Mettu*-4
- The bad accuracies may be the compression of the features using Histogram
- Future Work
  - The approach need to be applied and tested across all set of *Adavu*s and Dancers
  - Instead of using Histogram, we may directly use the feature dimension: 12x2x480x640 in SVM
  - We may use histogram as used in HoG

Research Plan

- Making the *bol* detection and *Sollukattu* recognition beater independent
- Posture and *Adavu* recognition was done on static Key postures. Motion-based Adavu recognition need to be done
- The following aspects associated with motion need to be explored.
  - Annotation of Motion
  - Motion classification: Non-Supervised Approach
  - Characterization, Modelling, and Recognition of the motion on the basis of the Trajectory
- Ontology for Motion Primitives

Publications

1. **Himadri BGS Bhuyan**, Partha P Das **Video Segmentation in Bharatanatyam Dance** in Expert systems with applications-2020 (Communicated)
2. **Himadri BGS Bhuyan**, Partha Pratim Das, and Mousam Roy. Motion classification in *Bharatanatyam* Dance Video in NCVPRIPG-2019, Springer, Singapore (Accepted).
3. Aich, Achyuta, Tanwi Mallick, **Himadri BGS Bhuyan**, Partha Pratim Das, and Arun Kumar Majumdar. NrityaGuru: A Dance Tutoring System for Bharatanatyam Using Kinect. "NrityaGuru: A Dance Tutoring System for Bharatanatyam Using Kinect." In National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics, pp. 481-493. Springer, Singapore, 2018.
4. Sankhla, Anindhya, Vinanti Kalangutkar, **Himadri BGS Bhuyan**, Tanwi Mallick, Vivek Nautiyal, Partha Pratim Das, and Arun Kumar Majumdar. Automated Translation of Human Postures from Kinect Data to Labanotation. In National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics, pp. 494-505. Springer, Singapore, 2018.
5. Tanwi Mallick, **Himadri BGS Bhuyan**, Partha Pratim Das, and Arun Kumar Majumdar. Research Data Set for Indian Classical Dance. Accepted for publication as an e-Book (Early skeletal version at: http://hci.cse.iitkgp.ac.in/).

Thank You

Name Of the Possible Upcoming Papers	**Written?**	**Communicated**
Video segmentation in Bharatanatyam Video	Yes	Yes
Adavu Recognition Using SVM and ED	Yes	No
Sollukattu Recognition using Bayesian N/W	Partially	No

More Work

If time permits then we may discuss the following (motion classification) work.

Introduction: Motion in *Bharatanatyam*

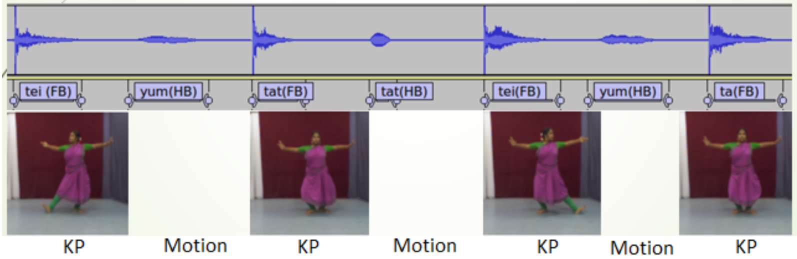


Figure: Occurrence of Motion and KPs

- A motion is a transition from one key posture to the next key posture
- Motions (M) and key postures (KP) occur alternately and may repeat in a performance
- Performance P consists of the interleaving sequence given by K1 M1 K2 M2 K3 M3 ... K(n-1) M(n-1) Kn.
- A motion comprises of several set of frames those are not momentarily stationary.

Motion classification: Motivation & Challenges

- Every dance form is characterized by a set of distinct motion sequences.
- Identifying and classifying the motions can lead to
  - Inter- and intra-ICD classification
  - Recognise the variance of performance within a particular dance form
  - Improve the quality of the applications like digital heritage and dance tutoring system
- Challenges
  - All the motions didn't have same no of frames even the similar ones, which makes implementation of ML model difficult.
  - Identifying a good similarity measure

Used Data Set

(a) *Used Data Set*

Adavu	# of frames in Performance			Total Motions	# of unique motions
	#1	#2	#3		
Natta-1	1546	1590	1532	32	4
Natta-2	1557	1522	1545	32	4
Natta-3	2680	2698	2760	64	8
Natta-4	5537	5531	5504	128	8
Natta-5	2580	2728	2748	64	10
Natta-6	2781	2764	2729	64	12
Natta-7	2828	3022	2706	64	14
Natta-8	2710	2811	2752	48	11
Overall	**22,219**	**22,666**	**22,276**	**496**	**71**

(b) *Sample Annotation*

KP ID	Start Frame	End Frame
N1D6B1M02	139	170
N1D6B2M01	186	218
....	....	....
....	....	....
....	....	....
N1D6B16M01	1436	1466

- *Adavu* Videos recorded using Kinect V1.0 in 30 fps.
- Recorded Data streams: Skeleton, RGB, Depth.
- We used RGB stream
- 496 motions are manually annotated

Work Flow

- **Input:** The motion frames in a given *Adavu* video
- **Output:** Labelled motions
- **Feature:** Optical flow vector Vx and Vy, Feature dimension: 480x640
- **Classifier:** kNN, The similarity measure computed using DTW
- Optical Flow is computed between each pair of frames in the motion.
- For a motion having 'N+1' Number of frames, the feature dimension: Nx2x480x640
- Since number of frames in each motion may not be same, DTW is used to compute cost between two motions.

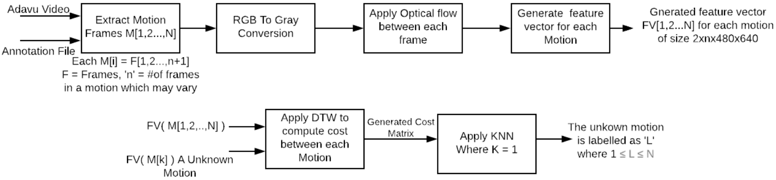


Figure: Motion classification Work Flow

Motion Classification Using KNN

- The motions involved in one set of *Adavu* of Dancer-6, Performance-1 are taken as unknown motion, and Performance-2 is tested.
- The motions having less than 10 frames are discarded and labelled as 'Noisy', since very less number of frames affects the Similarity measures and in turn affects the accuracy.
- By the removal of noisy motions, the accuracy of the Natta-6 to 8 increased.
- No noisy motions detected in Natta-1 to 5.

Adavu	Motion Accuracy (%)	
	With Noisy frames	Without Noisy frames
Natta-1	100	100
Natta-2	94	94
Natta-3	80	80
Natta-4	60.56	60.56
Natta-5	70	70
Natta-6	62	70
Natta-7	18	60
Natta-8	20	58.67

Table: Result

Result Analysis

- The best performing *Adavu*s are *Natta*-1, 2, 3 whereas the worst performing *Adavu*s are *Natta*-7 and 8 where accuracy hovers around 60% after removal of noisy frames. The possible Reasons of bad accuracy:
  - The larger difference of number of frames in the motions affects similarity measure.
    Example: M1 = 60, M2 = 11 and M = 12 frames. Now the cost of DTW(M1,M) $>$ DTW(M2,M), $\Rightarrow$ M=M2, but in reality M=M1.
  - Complexity of motions
  - Fast motions
  - Occlusion

Future Work:

- The Algorithm is to be tested across all the *Adavu*s.
- HOOF may be tried as a feature for DTW in KNN

The next slides: To understand the **Optical Flow** and **DTW**

Solving Optical Flow Equation

Let's solve the equation $I_x.u + I_y.v + I_t = 0$

$$\begin{aligned} I_x.u + I_y.v &= -I_t \\ \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} &= -I_t \end{aligned} \tag{1}$$

The above equation can be solved using least sqaure solution approach. If $A * d = b$ then $(A^T * A)d = A^T * b$. Now mapping this equation to equation-1

$$\begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_x \\ I_y \end{bmatrix} .I_t$$

$$\begin{bmatrix} {I_x}^2 & I_x.I_y \\ I_y.I_x & {I_y}^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_x.I_t \\ -I_y.I_t \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} {I_x}^2 & I_x.I_y \\ I_y.I_x & {I_y}^2 \end{bmatrix}^{-1} \begin{bmatrix} -I_x.I_t \\ -I_y.I_t \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} {I_{x_i}}^2 & \sum_{i=1}^{n} I_{x_i}.I_{y_i} \\ \sum_{i=1}^{n} I_{y_i}.I_{x_i} & \sum_{i=1}^{n} {I_{y_i}}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{n} I_{x_i}.I_{t_i} \\ -\sum_{i=1}^{n} I_{y_i}.I_{t_i} \end{bmatrix}$$

For patch of size $r \times c = n$, It's center pixel's (u, v) is computed by above equation

Example: Optical Flow Computation

let's consider two image patches of size 3×3 and compute (u, v) at center pixel. The require parameter I_x and I_y which is nothing but the derivative along X and Y axes respectively. Again I_t = Image difference or derivative along time axis.

1	2	2
2	1	1
1	2	1

(F_t)

2	1	2
1	1	1
1	2	1

(F_{t+1})

After Zero Padding $\Rightarrow$

1	2	2	0
2	1	1	0
1	2	1	0

(F_t)

2	1	2
1	1	1
1	2	1
0	0	0

(F_{t+1})

Table: Two Image frames (F_t & F_{t+1}) in time 't' and 't+1'

1	0	-2
-1	0	-1
1	1	-1

(I_x)

1	-1	-1
-1	1	0
-1	-2	-1

(I_y)

-1	1	0
1	0	0
0	0	0

(I_t)

Table: The Computed values of I_x, I_y and I_t

Compute the parameters ${I_x}^2 =$
$1^2 + 0^2 + (-2)^2 + (-1)^2 + 0^2 + (-1)^2 + 1^2 + 1^2 + (-1)^2 = 10$
Similarly ${I_y}^2 = 11$, $I_x.I_y = I_y.I_x = 2$, $I_x.I_t =$ -2, $I_y.I_t =$ -3

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 10 & 2 \\ 2 & 11 \end{bmatrix}^{-1} \begin{bmatrix} -2 \\ -3 \end{bmatrix} \Rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0.1509 \\ 0.2453 \end{bmatrix}$$

Example: Weighted Binning

Magnitude of 4 pixels M= {40, 8, 12, 120} and it's corresponding Angle/Direction A = {10, 40, 20, 25}

```
Assume the Given Bins as follows
[0,    0,    0,    0] -> Initial value is zero
 0th   20th  40th  60th -> Bin intervals

 Step-1: The angle = 10, magnitude = 40
 The angle 10 having an equal distance from 0th Bin and 20th Bin = |0-10| = |20-10|= 10
 So the magnitude = 40 equally divides between 0th and 20th Bins
 [0+20,    0+20,    0,    0] -> updated Bin values
 0th       20th     40th   60th -> Bin intervals

 Step-2: Angle = 40, Magnitude = 8
 The angle 40 finds its direct match with 40th Bin
 So, The Magnitude 8 goes into the 40th Bin
 [20,    20,    0+8,    0] -> updated Bin values
  0th    20th    40th   60th -> Bin intervals

  Step-3: Angle = 20, Magnitude = 12
  Here Angle 20, finds a match with 20th Bin
 [20,    20+12,    8,    0] -> updated Bin values
  0th     20th    40th   60th -> Bin intervals

  Step-4: Angle = 25, Magnitude = 120
  The angle 25, distance from 20th Bin = 5 and
  distance from 40th Bin = 15, so magnitude 120
  splits into 90+30. 90 unit goes to 20th Bin
  and 30 unit goes into the 40th Bin

  [10,    32+90,    8+30,    0] -> updated Bin values
  0th     20th      40th     60th -> Bin intervals
```

Illustration of DTW

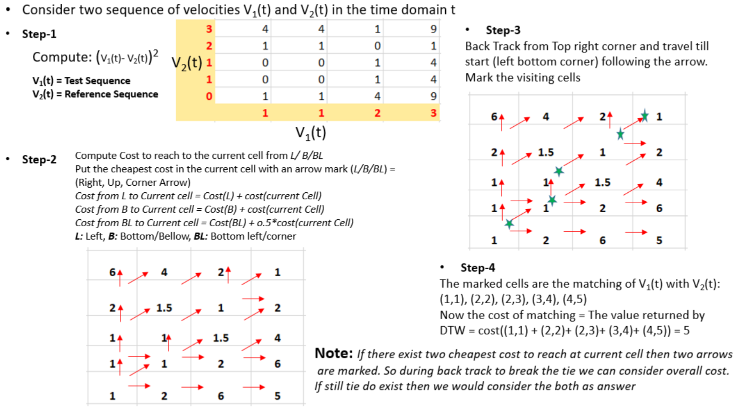


Figure: Illustration of DTW