Automated Analysis and Interpretation of Bharatanatyam Dance

Himadri B.G.S Bhuyan

Under the Supervision of
Prof. Partha Pratim Das

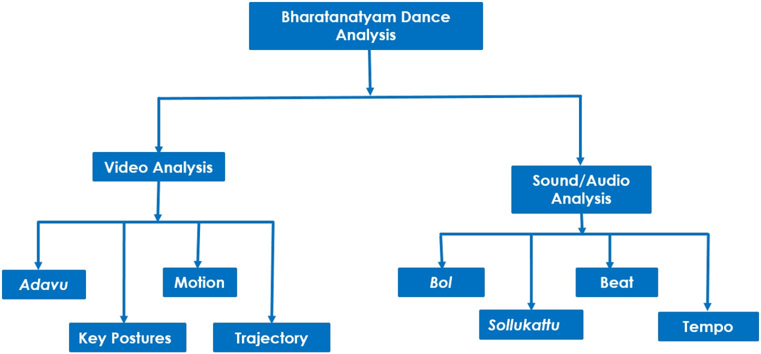
Rajendra Mishra School of Engineering Entrepreneurship
Indian Institute of Technology Kharagpur

Extension Seminar 2019

July 23, 2019

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 Done(KP), Audio Done	•Motion Annotation 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	• Recognition without splitting to the *Bol*s • Beater independent Recognition
Key Posture Recognition	•Feature: Angle & HOG •Recognizer: SVM and GMM	•To scale with the multiple Adavus •To Explore Bayesian techniques
Adavu Recognition Using KP	•HMM •SVM & Edit distance (ED) on Angle Feature	• SVM & ED on HOG Feature •*Adavu* Recognition Including Motion aspect
Applications	—*NrityaGuru*[4] —Human Postures to *Labanotation*[5]	•To be improved •To try on Dance • Automatic Annotation tool
Motion & key Frame Detection	•Non-adaptive & Rule based using Frame differencing & Bit-plane using RGB •Using velocity of skeleton Joints	• Adaptive & rule based (Non-ML) • ML Approach
Motion Classification	- -	• Using HoG/HOOF on RGB data • Using Trajectory of Limb joints

Table: Work Status

Recent Work

- Automation in detection of key frames & motion frames
  - Adaptive & Rule based Approach
  - ML approach
- *Adavu* Recognition
  - Feature: HoG, Recognizer: SVM & ED

Auto Detection of Key frame & Motion frame: Introduction

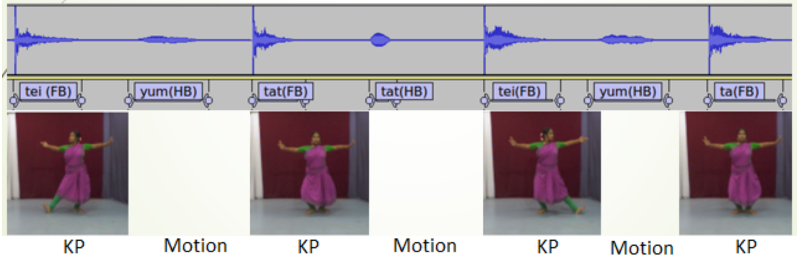


Figure: To understand Key frame and Motion frame occurrence

- 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.
- Key Frames (KFs): Momentarily stationary frames during occurrence of KP
- Motion frames (MFs): Frames during occurrence of Motions

Auto Detection of KF & MF: Earlier Works

- Work done by others[1]
  - Two Frame differencing (RGB Data). Accuracy (Avg): 74.14%
  - Velocity of limb joints (Skeleton data). Accuracy (Avg): 70.32%
- Our Initial approach
  - Non-adoptive threshold approach using Image differencing and bit-plane
  - Tried only on 12 videos
  - Accuracy (Avg): 90.14%
- Shortcomings
  - No Adaptive thresholds: For each video manual thresholds are defined
  - No automation to compute the threshold
  - Time consuming to detect proper threshold
- Scope
  - Scope of improvement
  - Adaptive threshold approach
  - Machine learning (ML) approach

[1] Characterization, Detection, and Synchronization of Audio-Video Events in Bharatanatyam Adavus, By T.Mallick, P.P.Das, 2018 Springer

Auto Detection of KF & MF: Objective & Current approaches

- Usage
  - To Distinguish Key posture and transitions in between automatically
  - To Recognize KP and *Adavu* sequences with auto detection of KF and MF
  - To build an annotation tool
- Current work
  - Adaptive & Rule based Approach (Non-ML)
  - ML approach
- Various Approaches

Technique	Input Image (RGB)			
	BGS Image		Non-BGS Image	
	ML	Non-ML	ML	Non-ML
Image diff + Bit-plane	✓	✓	✓	✓

Table: Various Approaches

BGS: Background Subtracted
Non-BGS: Without Background Subtracted (WBGS)

KF and MF Detection: Used data set

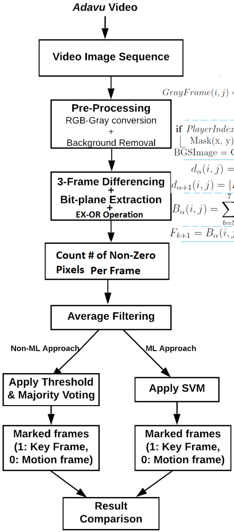
Adavu Name	Variations	# of Dancers	# of Videos
Joining	3	3	9
Kartari	1	3	3
Nattal	8	2	16
Tattal	5	3	15
Mandi	2	3	6
Mettu	4	3	12
Natta	8	3	24
Paikal	3	3	9
Pakka	2	3	6
Sarika	4	3	12
Sarikkal	3	3	9
Tatta	8	3	24
Tei-TeiDhatta	3	3	9
Tirmana	3	3	9
Utsanga	1	3	3
Total	58		166

Table: Data Set

- Video recorded using Kinect V1.0 in 30 fps.
- Recorded Data streams: Skeleton, RGB, Depth. We use RGB stream

KF and MF Detection: Process

Entire Process



$$GrayFrame(i,j) = 0.299 * Frame(i,j)_R + 0.587 * Frame(i,j)_G + 0.144 * Frame(i,j)_B$$

if *PlayerIndex(i, j) > 0* **then**
  Mask(x, y) = 1
BGSImage = GrayImage * Mask

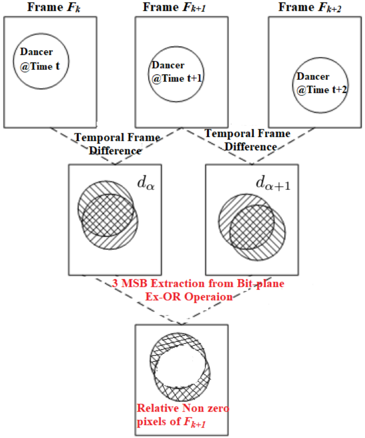
$$d_\alpha(i,j) = |F_{k+1}(i,j) - F_k(i,j)|$$

$$d_{\alpha+1}(i,j) = |F_{k+2}(i,j) - F_{k+1}(i,j)|$$

$$B_\alpha(i,j) = \sum_{b=5}^{7} 2^b * d_\alpha(i,j)_b$$

$$F_{k+1} = B_\alpha(i,j) \oplus B_{\alpha+1}(i,j)$$

Three Frame differencing & bit-plane extraction



KF and MF Detection: Significance of Ex-OR

Bit-Wise EXOR of all 8 Bits

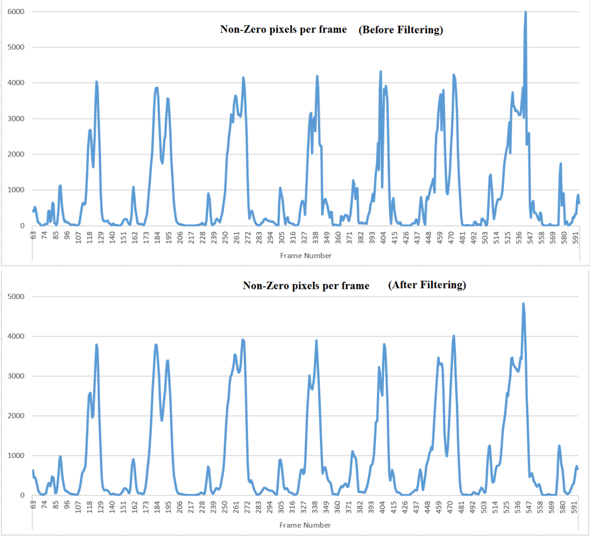
Figure-1

Bit-Wise EXOR of 3 MSBs

Figure-2

- EX-OR operation shows the changes in position of the dancer with time (Figure-1)
- 3 MSBs Ex-OR indicates only the moving pixels (Figure-2)

KF and MF Detection: Average Filtering



KF and MF Detection Non-ML approach deciding threshold

- Non-zero pixel counts per frame is the decider
- An adaptive threshold is must
  - Entire range of thresholds and its corresponding accuracy is analyzed

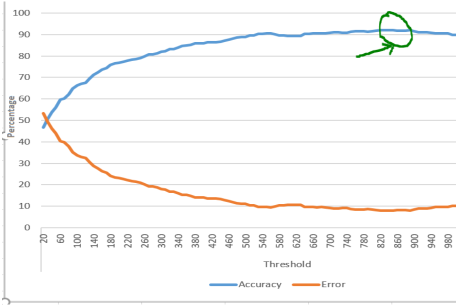


Figure: Example: Mettu-1 Dancer-3

- Outcome: $AdaptiveThreshold = \frac{\sum_{i=1}^{k} CountNonZeroPixels(F_k)}{k}$

KF and MF Detection: Non-ML Approach cont..

- The difference between Maximum possible accuracies and the accuracies using *AdaptiveThreshold* varies between 0.5% to 2.5%.

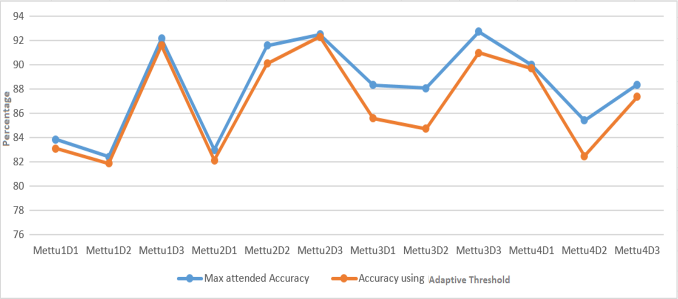


Figure: Example: Mettu Adavus

- $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$ TP: True +ve, TN: True -Ve

  FP: False +ve, FN: False -ve

KF and MF Detection Non-ML Approach: Majority Voting

- Majority voting to reduce FP & FN
- **Approach-1:** A sliding window acquiring three frames – check if there exist a key frame in between two motion frame and vice versa and the middle frame is updated.
- **Approach-2:** A sliding window of 5 frames – count the number of Key frames and Motion frames. As per the majority voting, the 3rd frame type is updated. Approach-2 Performs slightly better.

Figure: Example: Mettu Adavus (Approach-1 Vs Approach-2)

KF Detection Non-ML Approach: Result Analysis

Adavus	BGS (KF)			WBGS (KF)		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Joining	83.694	64.461	72.829	79.941	64.030	71.106
Kartari	98.539	49.080	65.524	97.095	50.193	66.176
Nattal	92.162	67.115	77.669	91.267	65.961	76.577
Tattal	85.614	57.071	68.488	84.872	58.465	69.236
Mandi	89.549	74.748	81.482	89.869	74.105	81.229
Mettu	93.165	87.732	90.367	93.226	86.747	89.870
Natta	85.519	82.562	84.014	85.105	82.298	83.678
Paikal	92.141	57.69	70.955	89.649	44.742	61.054
Pakka	77.847	43.328	55.671	74.536	41.240	56.493
Sarika	72.474	68.740	70.558	72.394	68.522	70.405
Sarikkal	89.247	65.432	75.506	90.490	63.829	74.856
Tatta	81.499	87.544	84.413	78.261	90.160	83.790
Tei-Dhatta	84.199	46.821	60.178	83.225	49.712	62.245
Tirmana	77.571	56.484	65.369	77.341	56.340	65.191
Utsanga	67.631	49.811	57.369	70.158	49.557	58.085
Average	84.723	63.908	72.026	83.829	63.044	70.978

Table: Non-ML Approach Result (in %) of (KF) Detection: WBGS Vs BGS

- $PrecisionKF = \frac{TP}{TP+FP}$, $RecallKF = \frac{TP}{TP+FN}$, $F1Score = 2 * \frac{Precision*Recall}{Precision+Recall}$
- By subtracting background, KF detection gives better result

MF Detection Non-ML Approach: Result Analysis

Adavus	BGS (KF)			WBGS (KF)		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Joining	69.515	85.343	76.620	70.664	83.296	76.462
Kartari	46.195	98.189	62.830	49.165	96.723	65.192
Nattal	67.999	91.781	78.120	67.930	91.519	77.980
Tattal	59.890	86.590	70.807	62.486	86.412	72.527
Mandi	72.888	87.993	79.731	71.572	88.012	78.945
Mettu	80.850	89.163	84.803	78.741	88.657	83.406
Natta	79.285	79.861	79.572	78.820	79.094	78.956
Paikal	51.781	97.857	67.725	58.024	97.799	72.835
Pakka	43.911	77.268	55.998	49.055	76.761	59.857
Sarika	58.362	61.915	60.086	57.824	61.257	59.491
Sarikkal	72.190	91.893	80.859	69.340	92.132	79.127
Tatta	73.178	63.293	67.877	80.473	60.564	69.114
Tei-TeiDhatta	62.726	93.016	74.925	67.625	92.634	78.178
Tirmana	59.216	79.085	67.723	59.778	79.364	68.193
Utsanga	72.469	84.798	78.150	70.456	85.347	77.190
Average	64.697	84.536	72.388	66.130	83.971	73.164

Table: Non-ML Approach Result (in %) of (MF) Detection: BGS Vs WBGS

- $PrecisionMF = \frac{TN}{TN+FN}$, $RecallKF = \frac{TN}{TN+FP}$, $F1Score = 2 * \frac{Precision*Recall}{Precision+Recall}$
- Results are not impressive, In most of the cases it is below 70%

KF & MF Detection: ML Approach

- ML doesn't require threshold for classification
- Features: The resultant pixel values generated for each frame by the Image Differencing and Bitplane extraction
- Feature length: 307,200 (= 480x640)
- Classifier: SVM

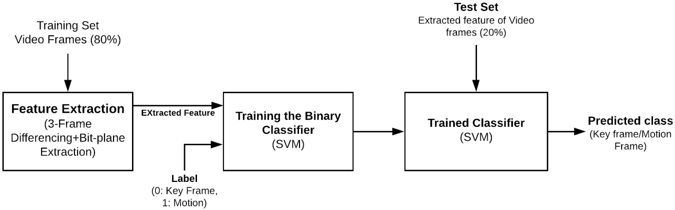


Figure: Training and Test Flow to classify *Key frame* & Motion frame using SVM

Extension

H. Bhuyan

KF Detection ML Approach: Result

Adavus	BGS (KF)			WBGS (KF)		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Joining	92.846	74.573	82.712	95.409	69.445	80.382
Kartari	95.118	68.297	79.506	88.424	47.895	62.135
Nattal	95.054	78.180	85.795	80.018	74.276	77.040
Tattal	92.092	85.034	88.423	87.673	84.359	85.984
Mandi	90.614	85.495	87.980	86.380	80.196	83.174
Mettu	95.670	85.416	90.253	97.171	90.656	93.800
Natta	94.485	86.256	90.183	94.188	84.810	89.253
Paikal	96.65	41.55	58.116	96.083	44.74	61.052
Pakka	74.266	64.458	69.015	68.268	58.930	63.256
Sarika	86.442	76.019	80.897	89.237	70.170	78.563
Sarikkal	91.447	67.812	77.876	95.682	69.189	80.307
Tatta	98.813	97.178	97.989	97.740	96.104	96.915
Tei-Dhatta	89.971	82.529	86.089	91.373	68.752	78.464
Tirmana	73.718	71.955	72.826	76.225	66.464	71.011
Utsanga	81.871	60.897	69.844	71.053	50.020	58.709
Average	89.937	75.043	81.167	87.662	70.440	77.336

Table: ML Approach: Result (in %) KF Detection: BGS Vs WBGS

- BGS performs much better than the WBGS
- Along with the precision the recall also improved, which reflects in F1 Score
- F1 score (Avg = 81.16%) is impressive and much more balanced than the Non-ML approach (Avg = 72.12%)

KF Detection Result comparison: Non-ML Vs ML Approach

- In ML with BGS most of the *Adavu* achieve more than or close to 90% precision.
- Average precisions are 84.72% and 89.93% in Non-ML and ML respectively.

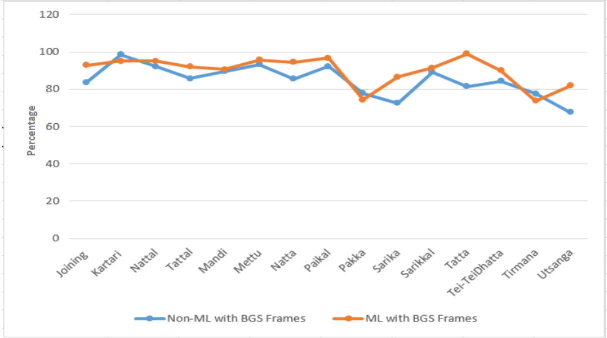


Figure: Comparison Non-ML BGS Vs ML BGS

- In few Adavus (Karatri, Pakka, Tirmana) Non-ML performs slightly better.
- Very slow motioned frame (= KF) some time detected as MF in ML where as in Non-ML that is being excluded by threshold.

MF Detection: ML Approach: Result

Adavus	BGS (KF)			WBGS (KF)		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Joining	77.829	94.210	85.239	70.143	95.370	80.834
Kartari	73.782	96.366	83.575	46.558	87.503	60.777
Nattal	76.302	82.752	79.396	70.108	91.272	79.302
Tattal	74.594	86.247	79.999	74.089	80.563	77.191
Mandi	85.569	90.324	87.882	79.456	85.468	82.353
Mettu	74.706	92.628	82.707	85.061	95.261	89.873
Natta	81.991	91.189	86.346	81.458	91.899	86.364
Paikal	74.149	96.592	83.896	51.397	94.168	66.499
Pakka	75.360	84.755	79.782	76.272	82.887	79.442
Sarika	66.372	81.008	72.963	51.694	77.136	61.903
Sarikkal	72.244	94.560	81.909	67.895	96.605	79.745
Tatta	93.219	97.106	95.123	90.289	94.668	92.427
Tei-Dhatta	92.784	95.976	94.353	81.316	96.665	88.329
Tirmana	80.007	82.603	81.284	72.575	81.978	76.990
Utsanga	78.576	91.400	84.504	71.035	86.148	77.865
Average	78.499	90.514	83.931	71.290	89.173	78.660

Table: ML Approach: Result (in %) MF Detection: BGS Vs WBGS

- BGS performs much better than the WBGS
- In most of the cases the precision is close to or above 75%
- Higher Recall value indicates FP $<$ FN; More Motion frames detected as key frames

MF Detection Result comparison: Non-ML vs ML

- In ML with BGS the average precision is 78.49% where as in Non-ML it is only 64.69%
- In Non-ML, the most of the *Adavu*s MF detection precision is under 70%, but in ML it is above 75%

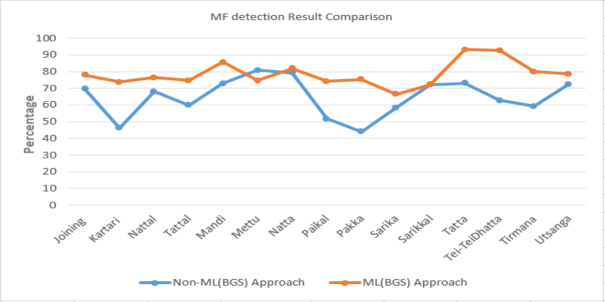


Figure: MF Detection Result comparison: Non-ML vs ML

- Except *Mettu Adavu*, in the rests, the ML approach performs better.
- In case of *Mettu*, in ML approach, slow motion frames detected as key frames are slightly higher than the Non-ML approach.

KF Detection ML approach Vs Image differencing approach[1]

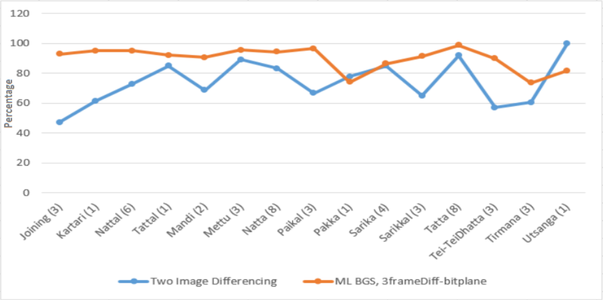


Figure: Comparison [1] Vs ML with BGS

- From figure, it is quite evident that our ML with BGS outclasses [1]
- The current approach gives an avg. accuracy of 89.93% where as in [1] it is only 74.14%
- The prev. approach only tried 10% of data where as we tried on 50% data set, but still the current approach performs better.

[1] Characterization, Detection, and Synchronization of Audio-Video Events in Bharatanatyam Adavus, By T.Mallick, P.P.Das, 2018 Springer

KF & MF Detection: Conclusion

- Major Contributions
  - Used bit-plane technique for dance – found to be very effective
  - Adaptive threshold is devised successfully for Non-ML approach
  - Explored ML technique
- Outcomes
  - ML approach gives consistently good result
  - The result of Non-ML is also not bad
  - Both the current approaches (ML/Non-ML) outperforms the earlier ones
- Future Scope
  - To use Depth Data in the same algorithm
  - To implement Histogram instead of a long feature (307,200 per frame) to improve the computational speed.
  - The Optical flow on RGB data to improve the MF detection accuracy.

Adavu Recognition on the basis of Key posture

- *Adavu*: The basic choreographic units of a dance sequence in *Bharatanatyam*
- Well defined sets of postures, gestures, movements and their transitions.
- It is Used to train the dancer.

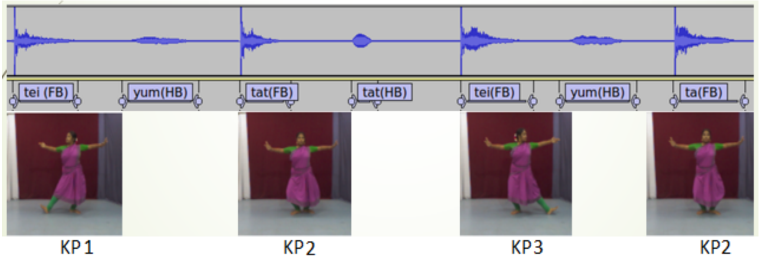


Figure: *Adavu* on the basis of KPs

Adavu Recognition on the basis of Key posture: Current Approach

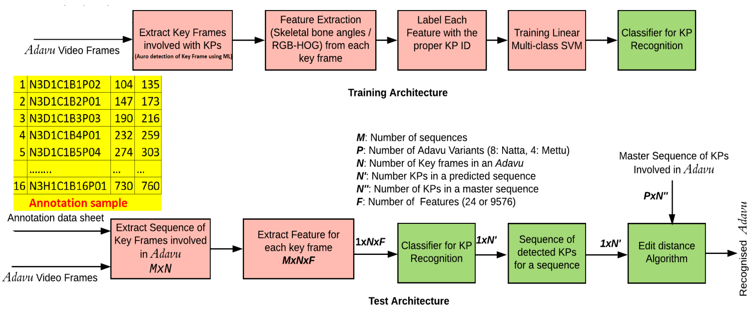


Figure: *Adavu* Recognition Using HoG Feature with Auto KF detection

- By looking to the Sample annotation file of *Natta Adavu*, The sample KP sequences which would represent an *Adavu* is as follows
  - Sequence-1: 104–147–190–232–274– .....–730
  - Sequence-2: 105–148–191–231–275– .....–731
- Each Frame in the *Adavu* sequence represents a KP

Adavu Recognition: KP Recognition Results

Adavu	# of Frames in Training Set	# of Frames in Test Set	Features	Recognizer	# of KP Classes	Accuracy (%)
Natta	7456	1865	Angle	SVM	23	97.10
Mettu	3663	908	Angle	SVM	32	87.66
Natta+Mettu	11119	2773	Angle	SVM	53	93.36
Natta	7456	1865	HOG	SVM	23	97.90
Mettu	3663	908	HOG	SVM	32	98.54
Natta+Mettu	11119	2773	HOG	SVM	53	94.15
Our Earlier work on KP						
–	–	–	RGB-D	GMM	23	82.84[1]
Natta	7381	1854	Angle	GMM	23	83.04[2]
Work done by others						

Table: Key Posture Recognition Results

- KP recognition result while considering HoG feature performs better

[1] Sharma, A. Recognizing Bharatanatyam Dance Sequences using RGB-D Data, HMM. PhD thesis, IIT Kanpur, 2013.

[2] Tanwi, A. Recognizing Bharatanatyam Dance Sequences using RGB-HOG, HMM. PhD thesis, IIT Kharagpur, 2017.

Adavu Recognition: Prev. Results Vs Current

Method	Feature	Adavu	# of Adavu	Test Sequences	Correct Recognition	Accuracy (%)
HMM [1]	RGB-D	–	12	–	–	80.55
HMM [2]	HOG	*Natta*	8	56	54	94.64
Work by others: without auto Key frame extraction						
SVM & ED	Angle	*Natta*	8	254	253	99.61
SVM & ED	Angle	*Mettu*	4	72	72	100.0
SVM & ED	Angle	*Natta* + *Mettu*	8 + 4	326	325	99.69
Our Earlier Work: without auto Key frame extraction						
SVM & ED	HoG	*Natta*	8	254	254	100.0
SVM & ED	HoG	*Mettu*	4	72	72	100.0
SVM & ED	HoG	*Natta* + *Mettu*	8 + 4	326	326	100.0
Our Current work: without auto Key frame extraction						
SVM & ED	HoG	*Natta*	8	254	252	99.21
SVM & ED	HoG	*Mettu*	4	72	72	100.00
SVM & ED	HoG	*Natta* + *Mettu*	8 + 4	326	324	99.38
Our Current work: with auto Key frame extraction						

Table: *Adavu* Recognition Results

- The current approach with auto *key frame* extraction gives slightly less accuracy, since it uses auto *key frame* extraction
- **Future Scope**
  - Increasing the data set and seeing the variation in accuracy
  - *Adavu* recognition using probabilistic model like Bayesian Network
  - *Sollukattu* associated with *Adavu* if can be checked, the possible *Adavu* will decrease for recognition.
  - Motion-based Adavu recognition

Research Plan

- Making the *bol* detection and *Sollukattu* recognition beater independent
- Recognition of *Sollukattu* without spliting into the *bol*s
- Trying out Depth data and Histogram technique in the current approach (Image Differencing and bit-plane) of KF and MF detection.
- 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 (Supervised & Non-Supervised Approach)
  - Characterization, Modeling, and Recognition of Trajectory
- Ontology for Motion Primitives

Publications

1. Aich, Achyuta, Tanwi Mallick, **Himadri BGS Bhuyan**, Partha Pratim Das, and Arun Kumar Majumdar. NrityaGuru: A Dance Tutoring System for Bharatanatyam Using Kinect. In Computer Vision, Pattern Recognition, Image Processing, and Graphics: 6th National Conference, NCVPRIPG 2017, Mandi, India, December 16-19, 2017, Revised Selected Papers 6, pp. 481-493. Springer Singapore, 2018. .
2. 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 Computer Vision, Pattern Recognition, Image Processing, and Graphics: 6th National Conference, NCVPRIPG 2017, Mandi, India, December 16-19, 2017, Revised Selected Papers 6, pp. 494-505. Springer Singapore, 2018.
3. Tanwi Mallick, **Himadri B G S 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/).

Name Of the Possible Upcoming Papers	Written?	Communicated
Adavu Recognition Using SVM and ED	Yes	No
Automation in detection of KF & MF	Yes	No
Sollukattu Recognition using Bayesian N/W	Partially	No

Thank You