In Proceedings of the 21st Int. Society for Music Information Retrieval Conf., October 2020

STRUCTURAL SEGMENTATION OF DHRUPAD VOCAL BANDISH AUDIO BASED ON TEMPO

Rohit M A **Vinutha T P** **Preeti Rao**
Department of Electrical Engineering
Indian Institute of Technology Bombay, India
{rohitma, vinutha, prao}@ee.iitb.ac.in

ABSTRACT

A Dhrupad vocal concert comprises a composition section that is interspersed with improvised episodes of increased rhythmic activity involving the interaction between the vocals and the percussion. Tracking the changing rhythmic density, in relation to the underlying metric tempo of the piece, thus facilitates the detection and labeling of the improvised sections in the concert structure. This work concerns the automatic detection of the musically relevant rhythmic densities as they change in time across the *bandish* (composition) performance. An annotated dataset of Dhrupad bandish concert sections is presented. We investigate a CNN-based system, trained to detect local tempo relationships, and follow it with temporal smoothing. We also employ audio source separation as a pre-processing step to the detection of the individual surface densities of the vocals and the percussion. This helps us obtain the complete musical description of the concert sections in terms of capturing the changing rhythmic interaction of the two performers.

1. INTRODUCTION

Dhrupad is one of the oldest forms of North Indian classical vocal music. A typical Dhrupad concert setting comprises a solo vocalist or vocalist duo as the lead and a pakhawaj player for the percussion accompaniment, with a tanpura in the background for the harmonic drone [1]. A Dhrupad performance lasts for over an hour and consists of an elaborate, unaccompanied *raga alap* followed by a composed piece, the *bandish*, performed along with the percussion instrument [2]. The bandish is not only presented as composed but also used as a means for further rhythmic improvisation (*laykari*), where the vocalist sings the syllables of the bandish text at various rhythmic densities and in different patterns [3, Chapter 10]. All the while, the pakhawaj accompaniment is either playing a basic pattern (*theka*) of the metric cycle (*tala*), a rhythmic improvisation to match the vocalist's improvisation, or a free solo improvisation while the vocalist presents the lines of fixed composition. The simultaneous rhythmic improvisation by both players is peculiar to the Dhrupad genre.

Figure 1 depicts the structure of a bandish performance from the vocalist's perspective. The intermediate refrain portions are the un-improvised sections where the artist sings a portion of the bandish before diving back into another spell of improvisation. A complete segmentation of

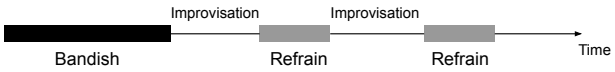


Figure 1: The structure of a bandish performance - vocalist's perspective [3]

a Dhrupad bandish performance would thus involve providing rhythmic descriptions of un-improvised and improvised sections pertaining to each - the vocals and the pakhawaj.

The goal of this work is to develop automatic methods for the structural segmentation of the Dhrupad bandish concert. With tempo and the relationships of the rhythmic densities of the individual instruments defining the distinct sections of a Dhrupad bandish concert, we explore new approaches to the reliable detection of these musical attributes as they vary across the concert. Given that vocal onsets are difficult to detect (even in isolated vocals due to the diversity inherent to singing), we turn to alternate methods for the direct estimation of the local rhythmic density. Advances in deep learning have led to the development of methods that treat the estimation of the predominant tempo from the raw audio spectral representation as a classification task [4–6]. We explore a similar approach for our task of estimating the changing surface tempo or rhythmic density across a concert audio. In view of the significant improvements reported in audio source separation in recent years, we also consider the use of source separation followed by tempo estimation for the constituent instruments in order to give a more complete description of each section.

The chief new contributions of our work are as follows: (i) a dataset of tempo markings and rhythmic density based structural segmentation annotations for Dhrupad bandish concerts, (ii) adapting a state-of-the-art tempo estimation method to the task of estimating local rhythmic density of the polyphonic mix, and (iii) the use of source separa-

tion to extend this to each instrument(vocals and pakhawaj) to eventually obtain a musically relevant segmentation of bandish concerts with section labels defined in terms of the rhythmic density inter-relationships.

2. BACKGROUND

Compositions in Hindustani music are sung at a tempo in one of roughly three broad ranges - *vilambit* (10.4-60 BPM), *madhya* (40-175 BPM) or *drut* (170-500 BPM) [3, p. 86]. This tempo is determined by the interval between the *matras* of the tala (a cyclic pattern of beats) that the composition is set to, and is referred to as the metric tempo. The metric tempo is fairly stable with only a gradual upward drift across the performance. However there are local variations in the rhythmic density of the singing or playing during what can be called episodes of improvisation that constitute the surface rhythmic density or surface tempo. For the voice, this is calculated using the number of syllables or distinct notes uttered in a unit interval and for the pakhawaj, the number of strokes played in a unit interval [3, p. 86], [7]. The surface tempo is found to generally be an integer multiple (ranging between 2 and 16) of the underlying metric tempo and we use the term 'surface tempo multiple'(*lay* ratio) to refer to this integer. The metric and surface tempi in this form of music thus have fairly objective definitions in terms of the performers' intentions and do not necessarily coincide with 'perceptual tempo'. And indeed as stated in [3, p. 85], the perceived tempo at extreme values of the metric or surface tempo may be quite different due to subdivisions at the lower end and grouping and accenting at the higher.

Related work on structural segmentation for Hindustani classical music can be found in [8–11]. The work in [8] relates to the segmentation of the initial unaccompanied alap portion of a Dhrupad vocal concert into the alap, jod and jhala sections. The methods exploit the changing nature of the energy, pulse clarity (salience), speed, and timbre of the vocals. In [10, 11], the task of segmenting the unaccompanied, and in [9] the accompanied portion of instrumental concert audios consisting of a lead melodic instrument(sitar, sarod) and a tabla accompaniment, was addressed. Signal processing methods based on finding onsets followed by periodicity detection were made use of for tempo and rhythmic density estimation. Section boundaries were obtained with the help of a similarity detection matrix, using frame-level ACF vectors of the detected onsets in [9], and using additional acoustic features and feature transformations in [11]. Faced with the problem of two instruments playing together, differences in the instrument timbres were exploited to separate the plucked string and tabla onsets in [9] to determine separately the metric and the surface tempo. Other source separation methods like HPSS [12, 13] and PLCA [14] have also been used to obtain tempo estimates for individual sources, which are then combined together to refine the overall tempo estimate.

In this work we address the structural segmentation of the bandish section in Dhrupad vocal performances, which has not yet been attempted. We propose to achieve this by first estimating the surface tempo using the CNN-based approach of [4] with a modified architecture to predict it directly as a multiple of the metric tempo. To obtain the surface tempo of each instrument, we make use of a pre-trained model provided by spleeter [15] that separates vocals from the accompaniment. We then detect section boundaries in a concert audio using changes in the estimated local surface tempi.

3. DATASET DESCRIPTION

To the best of our knowledge there is no existing dataset of tempo and segmentation related annotations for Dhrupad bandish performances. The dataset chosen for this work contains 14 concert audios in the *vilambit* and *madhya laya* - 8 from the Dunya corpus [16] and the rest from publicly available, good quality recordings. 9 of the 14 are by the vocalist duo Gundecha brothers, and the others by Uday Bhawalkar. Each recording is of a single bandish performance by the vocals, accompanied by pakhawaj, with a tanpura in the background. The recordings are 8-15 minutes long and the total duration of the dataset is about 3 hours. The performances are not all in the same raga or tala with at least one composition in each of 4 distinct talas commonly found in Dhrupad. 7 more publicly available audios were partially annotated to balance the cross-validation dataset described in Section 3.2.

3.1 Annotations

Annotations are of (i) the *sam* positions of the tala, i.e., the cycle boundaries, across the concert (ii) boundaries marking changes in the surface tempo multiple of each instrument and (iii) a label for each section in terms of the surface tempo multiple of each instrument. The annotations were marked by one of the authors, who is a trained musician, using the relatively objective criteria described here.

Information about the tala was obtained from the metadata accompanying the recording. With this, the sam positions were inferred either from the particular stroke of the pakhawaj or the syllable of the bandish refrain that appears on the sam in performance [17], or the number of matras elapsed since the previous sam. Although slight deviations are commonly observed in the metric tempo, large abrupt jumps do not occur. Hence, once a pass was made over the entire audio, the annotations were corrected at points of ambiguity to ensure coherence with adjacent sam markings. The metric tempo was then calculated versus time, once for every cycle, by dividing the cycle duration by the number of matras in the tala.

A section boundary was marked whenever the rhythmic density of either instrument changed and the new density was maintained for at least a duration of 5s. As mentioned earlier, the surface tempo is typically related to the metric tempo as an integer multiple. Therefore every section was labelled with the surface tempo multiple of each instrument, determined by calculating the rate of events (syllables for the vocals and strokes for the pakhawaj) as a mul-

tiple of the metric tempo in the section. Pauses at the pulse level occurring between syllables or strokes were considered as musical events contributing to the surface tempo, while pauses longer than 5s were labeled as having no surface tempo. A more detailed discussion on this appears in [7]. The maximum of the vocal and pakhawaj surface tempo multiples was then added to the section label as the net surface tempo multiple denoting the overall level of rhythmic density. Henceforth, we use the abbreviations m.t., s.t. and s.t.m. to refer to the metric tempo, surface tempo and surface tempo multiple.

Figure 2 is a visualisation of the annotations for a portion of a bandish audio in the dataset [1]. This roughly 4 minute long snippet captures a few sections - (a) vocal s.t. at 4 times the m.t.(∼60 BPM) and pakhawaj at 8, (b) vocals at the m.t. and pakhawaj at 16 times - in each of these the net is due to the pakhawaj, and (c) both at 4 times, where the net is due to both.

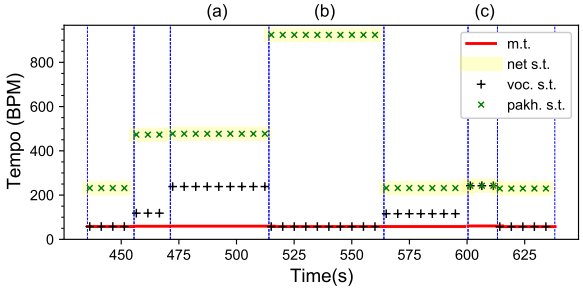


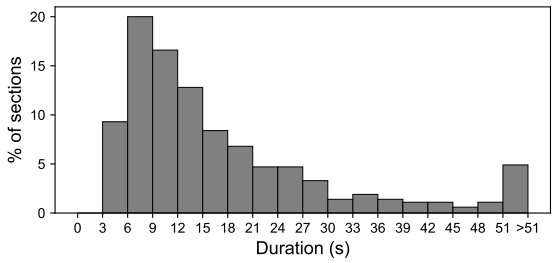
Figure 2: The ground truth metric tempo (m.t.) and surface tempo (s.t.) annotations for a portion of an audio in the dataset. Vertical dashed lines indicate section boundaries.

3.2 Dataset Statistics and Train-test Split

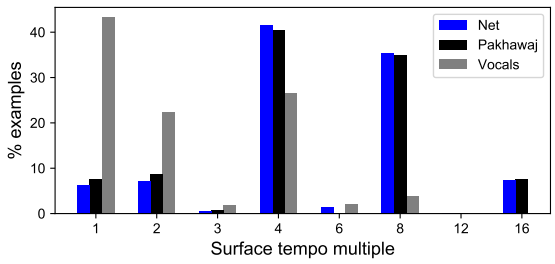
Every annotated section is homogenous in the sense that the s.t. of each instrument remains the same throughout its duration. We therefore pool the sections from all the concert audios into a dataset for training and testing our methods, treating each section as an independent entity. The total number of sections comes up to 634 (593 from the completely annotated and the rest from the partially annotated audios), but they are not all of similar durations. Figure 3 (a) shows the distribution of section durations with a single bar at the end for values more than 51s. We see that a section is mostly between 6 and 20s long. With the goal of tracking the s.t. as it is changing across a performance, we need to perform tempo estimation on shorter examples from each section. The duration of these examples is set to be 8s since a higher value would give us no examples from the large number of sections that are only 6-9s long. Further, for the slowest tempo in the dataset of about 30 BPM, an 8s duration would contain at most 4 beats, fewer than which may not be sufficient for accurate tempo estimation.

The distribution of s.t.m. in the dataset for each instrument and the net is shown in Figure 3 (b) in terms of the

[1] https://musicbrainz.org/recording/178b4cf6-88e6-414d-bfbd-3d90bb368a9a



(a)



(b)

Figure 3: Distributions of (a) section duration and (b) net, pakhawaj and vocal s.t.m. across our dataset of 1127 examples.

relative number of non-overlapping 8s examples (extracted from sections) available at each integer multiple, out of a total of 1127 examples. The dataset has a narrow m.t. range of 30 - 85 BPM, but the observed range of s.t. extends upto a large 960 BPM, due to the nature of the *lay* ratios. For the pakhawaj, we find that the multiples 4 and 8 are more abundant than 1, 2 and 16, while the multiples 3, 6 and 12 are nearly absent. For the vocals, 1, 2 and 4 are most represented and even though the multiples 3, 6 and 8 have a similar share, the sections for 8 were found to come from several concerts, while 3 and 6 were only found in a couple. We thus retain only the sections with s.t.m. values from the set {1, 2, 4, 8, 16}.

To manage the data imbalance, while generating the 8s training examples, the hop between consecutive examples is kept shorter for sections belonging to the less populous s.t.m values. We also augment the dataset by time-scaling the audio of each section [18] using one or more factors in the range {0.8, 0.84, 0.88, ... 1.2} (the s.t.m. label remains the same), generating more time-scaled versions for the less populous classes. The whole pool of examples is divided into three folds such that all the examples from a single audio section are assigned to the same fold, and each fold has a similar distribution of the s.t.m. values.

4. METHODS

We consider the recent CNN-based tempo estimation method from [4] (denoted as tempo-cnn) for our work. After examining the viability of the pre-trained model, we first attempt to train new models with the same architecture on our dataset, and then propose some suitable modifications.

4.1 Metric Tempo Estimation

The m.t. of a Dhrupad bandish performance gradually drifts across a performance. Hence, we are interested in estimating it locally and tracking it versus time. With the m.t. range of our dataset being a subset of the tempo-cnn output range, the pre-trained model can be used as it is to observe the nature of its predictions. Upon obtaining estimates frame-wise at 0.5s hops and picking the output class with the highest confidence in each frame, it is found that the model almost always makes octave errors, which is to be expected since the m.t. in our case is not always the perceptual tempo that the model was trained to estimate. We fix these errors by constraining the predicted tempo to lie in the range of m.t. values in the dataset.

We do not attempt to train a new model for m.t. estimation and instead compare the above with a non-learning based approach from [9]. A spectral flux based method is used to obtain the onsets and the autocorrelation function is calculated on 12s long windows at 0.5s hops for values of lag upto 2s. The tempo candidates are constrained to be in the required range and an additional Viterbi smoothing step is used to penalise jumps and obtain a consistent estimate across a concert. We refer to this as the odf-acf method. We also note that the metrical cycle tracking work of [19] offers an alternative that can be investigated for m.t. estimation in future work.

4.2 Surface Tempo Estimation

The s.t. values in our dataset fall outside the tempo-cnn output range. And since the task requires correct identification of tempo without octave errors, using the pre-trained tempo-cnn is not possible. If we are to re-train tempo-cnn on our dataset by increasing the output range, the huge size of the range presents a problem due to the resulting target class imbalance. Therefore, given that the s.t.m. is one of a small set of integer values, we modify the task to predicting this multiple instead of the actual s.t. value.

An attempt to train new models using the tempo-cnn architecture on our dataset by reducing the final softmax layer dimensions does not turn out to be fruitful as the model overfits due to its high capacity and the small size of our dataset. The main issues seem to be the high number of dense layers at the end and the large filter lengths in the multi-filter modules. After a series of simplifications with some inspiration from [5], the architecture summarised in Table 1 is found to be promising (details in [7]). The reduction of dense layers and the addition of dropout layers is found to be crucial in overcoming overfitting. To prevent too much information from getting cut-off due to the dropout, the p value is set to 0.1 in the first three conv. layers, and 0.5 in the later ones. As for the multi-filter conv. layer, fewer filters in parallel and smaller filter lengths are found to make the network easier to train. However, to ensure adequate capacity, the number of filters in each layer is kept moderately high.

Every 8s training example is transformed to a log-scaled mel-filtered magnitude spectrogram, using the following parameters - 40ms windows, 20ms hops and 40 mel filters over the band 20-8000 Hz, at a sampling rate of 16kHz. The input to the network is a spectrogram of size 40 x 400 with the values normalized to lie in the range 0 - 1, and the target is one of 5 classes corresponding to the 5 s.t.m. values - 1,2,4,8,16. The network is trained using CCE loss on examples from two folds, with the other fold as the validation set, for a maximum of 500 epochs. Training is carried out using the Adam optimizer with a learning rate of 1e-4 and a batch size of 32, and is halted early if the validation loss does not decrease for 50 epochs.

Layer	Dimensions
Input	40 x 400
(BN, Conv, ELU, DO) x3	16 x 1 x 5
AvgPool	5 x 1
BN, MF Conv, DO	12x {1x16, 1x32, 1x64, 1x96}
Concat, Conv	16 x 1 x 1
AvgPool	1 x 400
BN, DO, FC, Softmax	# output classes

Table 1: Proposed model architecture, adapted from [4] & [5]

4.3 Extension to Separated Sources

Given our interest in estimating the s.t. of each instrument to obtain a more complete rhythmic description and the section boundaries in a concert, the pre-trained 2-stems model by spleeter [15] is used to separate the mixture audios into vocals and accompaniment, and new models with the same architecture as proposed above are trained to predict the s.t.m. for each. The dataset of sections remains the same but the input examples are of the separated sources and the training and validation folds are generated again for each source to balance the number of examples across the corresponding classes. The target classes for the pakhawaj are the same as earlier but those for vocals do not include the s.t.m. 16.

4.4 Boundary Detection and Section Labelling

We aim to automatically identify sections in a concert by looking for abrupt changes in the s.t.m. values of each instrument across the concert duration. For this task only the completely annotated 14 concert audios are used. Estimates of s.t.m. are obtained once every 0.5s using 8s long excerpts over the entire duration of each audio. While doing so, each excerpt is presented to that saved model out of the three from the 3-fold CV procedure, to which no portion of the section that this excerpt lies in was presented as a training example, thus preventing any train-test leak. The output class with the highest confidence is taken as the s.t.m. estimate. This procedure is applied to the mixture and the source separated audios. A boundary is marked wherever the s.t.m. of either instrument changes, and the section label is the tuple of the three s.t.m. estimates.

We experiment with two methods for obtaining the three s.t.m. estimates. One, the three values are estimated

Method	Accuracy 1	Accuracy 2
tempo-cnn	5.2	73.8
tempo-cnn with range constraint	71.6	74.7
odf-acf	72.0	72.0

Table 2: Metric tempo estimation accuracies (%) at 4% tolerance using tempo-cnn [4] and the odf-acf method [9].

independently, and we refer to this method as `seg1`. Here the net s.t.m. may not be equal to the higher of the other two (which should be true by definition). We thus report results using the model output for the net s.t.m. as well as by simply taking the maximum of the other two as the net s.t.m. value. Two, to investigate whether using the expected relationship between the three s.t.m. values helps improve performance, instead of obtaining them independently, we pick that tuple of the three estimates in every frame which has the highest average classifier confidence value and in which the net s.t.m. is the maximum of the other two. We refer to this method as `seg2`. To reduce the number of false alarms, a post-processing step is used with each method to smooth the outputs by constraining the duration of a detected section to be at least 5s. This is implemented by removing the boundaries of any section that is shorter and replacing the label by that of the previous section.

5. EXPERIMENTS AND RESULTS

5.1 Metric Tempo Estimation

To evaluate m.t. estimation we calculate *accuracy1* and *accuracy2* (allowing for octave errors) with a tolerance of 4% across each audio at a 0.5s frame-level and then average it across the dataset. We find that both the methods fare equally well (Table 2) and the simple fix of including a range constraint significantly improves *accuracy1* for tempo-cnn (except in cases where the prediction is an octave off but already in the m.t. range).

A closer look at concert-wise scores revealed that the accuracy was below 70% in the same 4 (out of 14) concerts in both the methods, where most of the errors were due to the predicted value being either 1.5 or 0.75 times the actual m.t. value. The tempo-cnn makes errors only in small portions of such concerts, but in the odf-acf method, due to the imposed penalty on jumps, the predicted tempo was found to be incorrect over longer durations. Even so, what we take away from the overall results is that for most of the concerts, m.t. is estimated well across the entire duration despite the presence of sections where both the instruments are improvising and playing at different multiples of the m.t.

5.2 Surface Tempo Estimation

Here, we first report the average 3-fold cross-validation accuracy values. This accuracy measures the proportion of 8s examples for which the s.t.m. was correctly identified. Table 3 shows the results for all three cases - estimation of net s.t.m. from the original mixture, and that of the individual instruments from separated audios.

Case	Net s.t.m	Vocal s.t.m	Pakhawaj s.t.m
Accuracy	75.2	69.1	76.9

Table 3: Average 3-fold cross-validation accuracies (%) for surface tempo multiple estimation

The results are poorer for separated vocals and better for pakhawaj, which reflects also in the net score, given that the net s.t.m is dominated by that of the pakhawaj. The class-wise performance is shown using a confusion matrix for each case in Table 4. In the case of vocals, classes 1 and 8 are estimated more accurately. For class 8, this could be due to the distinct nature of vocalisation and the limited diversity of examples due to fewer available sections. For class 1, most examples come from sections where the bandish is sung at a steady rate without improvisation thus making tempo estimation easier. For class 2, sections often come from the earlier stages of improvisation in a concert where the singing is not fully rhythmic and is characterized by pauses, melismatic singing and changes to other s.t.m. levels, making the estimation harder. The confusions between classes 2 and 4 could also be due to some bleed of pakahwaj into the vocals during source separation.

In the case of net and pakhawaj s.t.m., classes 1 and 2 are estimated quite accurately, while the other classes are confused with their immediate neighbours. The class 16 being confused with 8 is most likely because of the presence of accents on every other stroke. We also notice a drop in the performance of this class in the case of separated pakhawaj when compared to the mixture audios, possibly due to a further loss of weak onsets after separation.

5.3 Boundary Detection and Section Labelling

We evaluate boundary retrieval performance using precision, recall and F-score (Table 5a). A predicted boundary is declared a hit if it falls within a certain duration of an unmatched ground truth boundary, and a false alarm otherwise. Results are reported at two values of temporal tolerance: $\pm$1.5s and $\pm$3s. The latter value is as used in [20] and the former is included with the reason that since a large number of sections are 6-9s long, even if both the detected boundaries are off by 1.5s, the detected section still captures at least half of the ground truth section.

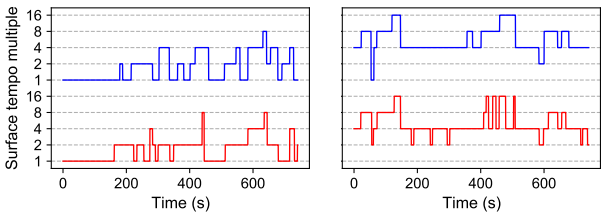
To evaluate section labelling, we report labelling accuracy (Table 5b) as the fraction of the duration of each concert that is correctly labelled (excluding regions where the ground truth is not one of {1,2,4,8,16}), averaged across the dataset, as defined in [21]. Each of the three s.t.m. labels are first evaluated individually and also when taken together (i.e., a frame is said to be correctly labelled only if all three labels are correct). We expect these scores to be different from the cross-validation accuracies reported in Table 3 as the test set is now no longer balanced, with

			Predicted				
			1	2	4	8	16
(a)	Ground truth	1	**90.1**	2.2	6.4	0.0	1.3
		2	5.8	**82.0**	**10.8**	0.8	0.5
		4	4.5	**13.4**	**66.9**	**11.9**	3.3
		8	2.4	1.8	**14.4**	**65.7**	**15.7**
		16	1.8	1.0	6.5	**15.1**	**75.5**
(b)	Ground truth	1	**77.3**	**15.5**	5.3	2.0	
		2	**21.0**	**50.8**	**26.2**	2.0	
		4	5.8	**20.1**	**64.6**	9.4	
		8	1.8	0.0	**13.2**	**84.9**	
(c)	Ground truth	1	**93.0**	0.9	5.0	0.8	0.2
		2	0.2	**83.3**	**14.4**	1.7	0.3
		4	5.8	**15.1**	**65.4**	**11.1**	2.6
		8	2.9	1.1	**11.5**	**69.9**	**14.5**
		16	1.4	1.1	6.1	**24.8**	**66.6**

Table 4: Confusion matrix of (a) net, (b) vocal, and (c) pakhawaj s.t.m. predictions (values in %)

the confused classes being the more common ones.

The individual labelling accuracies are quite similar for the pakhawaj and net tempo labels, slightly lower for the vocals, but much lower for getting all the labels right in every frame. With `seg1`, we see that the vocal and pakhawaj estimates are reliable enough that taking their maximum as the net s.t.m. instead of using the model estimate improves the net s.t.m. labelling accuracy. Hence, for the evaluation in the last column, the net is taken as the maximum of the other two. Although this seemingly renders the model trained to predict the net s.t.m. not very useful, we see in `seg2` that using it to obtain all the estimates together improves all the accuracies, proving its utility.



(a) (b)

Figure 4: The ground truth (above) and estimated (below) s.t.m. labels of the (a) vocals and (b) pakhawaj across the concert *GB_AhirBhrv_Choutal*.

Although better tempo estimation should result in better boundary detection since the boundaries are based entirely on tempo changes, the boundary detection results using `seg2` are only slightly better than `seg1`. In both the cases, the smoothing step was found to improve the results (detailed in [7]). Looking at the vocal and pakhawaj s.t.m. estimates obtained using `seg2` in Figure 4, we see that for both the instruments, at a coarse level, the various surface tempo regions are captured well. And while for the pakhawaj, finer section changes are also estimated accurately, such changes are not tracked well in the case of vocals, thus reducing the overall boundary detection scores.

	±1.5s tolerance			±3s tolerance		
	Prec.	Rec.	F-sc.	Prec.	Rec.	F-sc.
`seg1`	0.27	0.38	0.32	0.39	0.54	0.45
`seg2`	0.29	0.38	0.33	0.40	0.53	0.45

(a)

	Vocals	Pakhawaj	Net from model	Net as max.	All 3 labels
`seg1`	67.2	68.7	66.9	67.5	45.9
`seg2`	67.7	71.0	70.4	-	48.6

(b)

Table 5: (a) Boundary detection performance and (b) s.t.m. labelling accuracies (in %).

6. CONCLUSIONS

We have presented a system that provides a complete rhythmic description of a Dhrupad bandish performance, enabling its segmentation into musicologically relevant sections based on the rhythmic interaction between the instruments. The metric tempo is estimated by adapting existing methods whereas the surface tempo, with its much larger dynamic range, is estimated in a novel manner by predicting its relationship with the m.t. to directly obtain the musically significant *lay* ratio. Because of the challenges presented by imperfect source separation, we benefit from using a model trained also on the mixture audios. We find that s.t.m. values at the lower and higher extremes are estimated better than the intermediate values. This, despite the intermediate values being the more represented classes in the dataset, points to the diversity in the acoustic realisations of the different surface densities. Future work could involve extending the dataset to encompass more singers and compositions in the *drut* lay, where we might see the same s.t.m. manifesting completely different acoustic properties. In such a scenario, estimating m.t. could help provide useful 'conditioning', and ways to jointly estimate the metric and surface tempi could be explored. Source separation can be improved by introducing new loss functions that preserve onsets better and hence allow better tempo estimation on separated audios. Finally this work provides an example of adapting available MIR methods to music genre specific problems.

Supplementary material

All the dataset details, annotations, code and pre-trained models are available here: `https://github.com/DAP-Lab/dhrupad-bandish-segmentation`.

7. REFERENCES

[1] B. C. Wade, *Music in India: The classical traditions*. Englewood Cliffs, New Jersey: Prentice-Hall, 1979.

[2] R. Widdess, "Involving the performers in transcription and analysis: a collaborative approach to Dhrupad," *Ethnomusicology*, vol. 38, no. 1, pp. 59–79, Winter 1994.

[3] M. Clayton, *Time in Indian Music: Rhythm, Metre, and Form in North Indian Rāg Performance*. Oxford, England: Oxford University Press, 2000.

[4] H. Schreiber and M. Müller, "A single-step approach to musical tempo estimation using a convolutional neural network," in *Proc. of the 19th Int. Society for Music Information Retrieval Conf.*, Paris, France, 2018, pp. 98–105.

[5] H. Schreiber and M. Müller, "Musical tempo and key estimation using convolutional neural networks with directional filters," in *Proc. of the 16th Sound and Music Computing Conf.*, Malaga, Spain, 2019, pp. 47–54.

[6] S. Böck, M. E. P. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," in *Proc. of the 20th Int. Society for Music Information Retrieval Conf.*, Delft, The Netherlands, 2019, pp. 486–493.

[7] M. A. Rohit and P. Rao, "Structure and automatic segmentation of Dhrupad vocal bandish audio," Unpublished technical report, arXiv:2008.00756 [eess.AS], 2020.

[8] P. Rao, T. P. Vinutha, and M. A. Rohit, "Structural segmentation of alap in Dhrupad vocal concerts," *Transactions of the Int. Society for Music Information Retrieval, Under review*, 2020.

[9] T. P. Vinutha, S. Sankagiri, K. K. Ganguli, and P. Rao, "Structural segmentation and visualization of sitar and sarod concert audio," in *Proc. of the 17th Int. Society for Music Information Retrieval Conf.*, New York City, USA, 2016, pp. 232–238.

[10] T. P. Vinutha, S. Sankagiri, and P. Rao, "Reliable tempo detection for structural segmentation in sarod concerts," in *Proc. of the 22nd National Conf. on Communications*, Guwahati, India, 2016, pp. 1–6.

[11] P. Verma, T. Vinutha, P. Pandit, and P. Rao, "Structural segmentation of hindustani concert audio with posterior features," in *Proc. of the 40th IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Brisbane, Australia, 2015, pp. 136–140.

[12] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stajylakis, "Music tempo estimation and beat tracking by applying source separation and metrical relations," in *Proc. of the 37th IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 2012, pp. 421–424.

[13] A. Elowsson, A. Friberg, G. Madison, and J. Paulin, "Modelling the speed of music using features from harmonic/percussive separated audio," in *Proc. of the 14th Int. Society for Music Information Retrieval Conf.*, Curitiba, Brazil, 2013, pp. 481–486.

[14] P. Chordia and A. Rae, "Using source separation to improve tempo detection," in *Proc. of the 10th Int. Society for Music Information Retrieval Conf.*, Utrecht, The Netherlands, 2009, pp. 183–188.

[15] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020, Deezer Research.

[16] A. Srinivasamurthy, G. K. Koduri, S. Gulati, V. Ishwar, and X. Serra, "Corpora for music information research in Indian art music," in *Proc. of the 40th Int. Computer Music Conf. / 11th Sound and Music Computing Conf.*, Athens, Greece, 2014, pp. 1029–1036.

[17] J. C. Ross, T. P. Vinutha, and P. Rao, "Detecting melodic motifs from audio for Hindustani classical music," in *Proc. of the 13th Int. Society for Music Information Retrieval Conf.*, Porto, Portugal, 2012, pp. 193–198.

[18] Rubber Band Library, "Rubber band library v1.8.2," https://breakfastquay.com/rubberband/, 2018.

[19] A. Srinivasamurthy and X. Serra, "A supervised approach to hierarchical metrical cycle tracking from audio music recordings," in *Proc. of the 39th IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Florence, Italy, 2014, pp. 5217–5221.

[20] K. Ullrich, J. Schlüter, and T. Grill, "Boundary detection in music structure analysis using convolutional neural networks," in *Proc. of the 15th Int. Society for Music Information Retrieval Conf.*, Taipei, Taiwan, 2014, pp. 417–422.

[21] J. Paulus and A. Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, August 2009.