Automatic Transcription of “Gambang” Balinese Gamelan

Sarah Lecompte-Bergeron
Faculté de musique
Université de Montréal
sarah.lecompte-bergeron@umontreal.ca

Dominic Thibault
Faculté de musique
Université de Montréal
dominic.thibault@umontreal.ca

I Putu Arya Deva Suryanegara
Faculté de musique
Université de Montréal
i.putu.arya.deva.suryanegara@umontreal.ca

Abstract

This article describes the automatic transcription process of a *gambang* ensemble, a type of Balinese gamelan, from multitrack audio recordings. This project employs artificial intelligence techniques that have proven efficient in the literature, namely multi-layer perceptron classifiers trained on chroma analysis, and here adapted to repertoire that offers specific challenges due to its musical realities. After describing the musical formation and its musical context, we will describe the method used for data acquisition, the analysis process of aforesaid dataset and transcription, concluding with an evaluation of the process' efficiency and discussion on further developments of the algorithm.

1 Context

1.1 *Pupuh gambang* [1]

Gambang is a type of gamelan percussion ensemble performed in Bali, Indonesia. This 7-tone gamelan is usually constituted of metallophones called ***gangsa*** and keyed bamboo instruments called ***cungklik***. *Pupuh gambang* is a type of repertoire found across the island which share the characteristics of notation on palm leaf manuscript. This traditional notation presents a series of single tones in Balinese writing (***aksara Bali***; Figure 1), giving insight on the “melodic” progression.[1] The relationship between this notation and what is executed by the musicians in performance is not explicit; it also relies on oral tradition and improvisation. In *Pupuh Gambang: Manuscript, Melody, and Music*, Jonathan Adams describes the idiomatic improvisation of *pupuh gambang* practice as “generative processes” where “[i]nputs consist of the *pupuh*, pre-established performance conventions, and [...] certain environmental cues [, and where] the output is the patterned sounds that instrumentalists play” [1].

We examine if we can devise an efficient transcription process of the music executed by musicians of *gambang* Balinese gamelan. In the following pages, we will detail the method used for data acquisition, the analysis process of aforesaid dataset and transcription, concluding with an evaluation of the process' efficiency and discussion on further developments.

1.2 Transcription

Adams already proposes a type of cipher notation (numbers, or Arabic numerals, representing note values played on instruments; Fig. 1) to represent *gambang* performances which he manually transcribed from audiovisual recordings [2]. Previously developed Automatic Music Transcription (AMT) software [3] take a waveform as an input and output a representation useful for the Western musician. We think the development of a tool offering automatic transcription in cipher notation of *gambang* sound files could significantly complement the traditional notation in the analysis of this rich repertoire by making the process of transcription much more efficient. This opens the possibility

[1] Adams [1] notes that the word melody “has connotations that are misleading” but is practical in our context.

to treat a larger corpus of various interpretations, which could shed light on regional and individual specificities of the musical genre [4]. In addition, the digitization of a large corpus could be the basis for generating musical material algorithmically, using, for instance, machine learning. Numbers represent tones in our system (Fig. 2), regardless of register.

Figure 1: An excerpt of the piece *Semarandana*, manually transcribed by the authors in *gambang* notation using *aksara Bali* (textured highlight), western notation, and cipher notation.

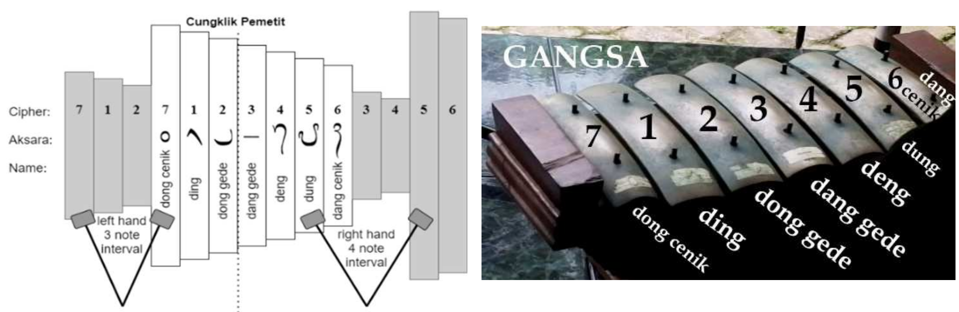

Figure 2: Correlation between representations of tones on a Kerobokan village version of a *cungklik pemetit* and *gangsa* as: cipher, *aksara Bali* for *gambang* notation, and note name[2].

In this study, our focus will be on formal sections where *cungklik* instruments perform interlocking patterns (*kilitan*[3]) [5]. These sections are particularly challenging for manual transcription due to the difficulty of source separation in such a dense and homogenic texture: *cungklik* instruments tend to blend because of their similar timbre, register, and dynamics. The resulting steady pulse perceived in such interlocking sections is called ***ketukan*** [6] (Fig. 1, one box or one 16th note is one *ketukan*) and it will be the base time unit of our transcription.

1.3 Data acquisition

To be able to perform the automatic transcription, our tool requires audio recordings from a performing ensemble. Multitrack recording of individual instruments comes as a solution for isolating similarly sounding *cungklik*. The recording procedure established requires that one microphone be placed over each instrument. Sanggar Seni Naradha Gita, an ensemble based in Kerobokan (Badung), played and recorded two interpretations of the piece *Panji Marga* and two interpretations of *Semarandana* in November 2021. The resulting recordings are used as a corpus

---

[2] According to the sequence of notes established by the disposition of the keys on the *gangsa* instruments, where the lowest note (1) is *ding* and the highest note (7) is *dong cenik* [5].

[3] Audio and video: https://github.com/AtrashDingDong/GaKo-alat/tree/main/gambangAMT

to develop and test our transcription algorithm. In a standard digital audio workstation (DAW), we verify that all sound files are synchronized. The four *cungklik* tracks are superimposed to create a master mix. This mix and the five individual tracks are exported separately to the .wav format. Each file is carefully named according to its content.[4]

2 Analysis process

For the analysis process, the graphical programming environment Max/MSP [7] is used along with the Fluid Corpus Manipulation (FluCoMa) library [8], which provides signal decomposition and machine learning toolsets. The following steps describe the analysis process carried out by our tool.

2.1 Importing previously described sound files into buffers

The sound content of each individual instrument is separated into harmonic and stochastic content using the FluCoMa object [fluid.bufsines~]. Only the harmonic content will later be analyzed.

2.2 Onset detection

The [fluid.bufonsetslice~] object detects attacks and outputs their location. The [fluid.waveform~] object enables visualization of the waveform and the associated detected onsets. Onsets can be adjusted, added, and/or removed with the mouse; or the detection parameters can be adjusted and the process retriggered. This data is stored in dictionaries [dict] for each sound file.

2.3 Onset matching

Correlation between individual instrument onsets and mix onsets is performed in order to quantize instrument strokes on a *ketukan* grid. For each individual instrument onset, we query the mix onsets dictionary for possible corresponding onsets. A window of deviation, adjustable by the user, is applied to allow for differences in time (here 0.2 s) between the onsets detected in the mix and the instrument stroke. If a match is found, the strokes are considered simultaneous and the location in the buffer is added to the instrument dictionary with the corresponding mix onset as the key. An error is flagged if no corresponding onset is found, and the user can adjust onset markers accordingly from the waveform display. Matched onsets are organized in a centralized dictionary.

2.4 Analysis, model training, and prediction

For each instrument onset identified, we want to determine which note has been played [9]. To do this, we will ask multi-layer perceptron (MLP) [10] models to predict tone values from a chroma analysis of the audio.

2.4.1 Chroma analysis

After the audio has been segmented, we are interested in analyzing its slices' chroma [11]. Chroma is an audio analysis that models the composition of pitch classes in a sound [8]. Note that an analysis of chroma is chosen here because *cungklik* instruments play at least two notes at the same time at an interval of one octave due to the doubling nature of the mallets (Fig. 2). Pitch value, sensible to discrepancies in range, would not be of use in this multilayered context. For each individual onset a user specifies, a 200 ms long slice of the resynthesized harmonic content is analyzed using the [fluid.bufchroma~] object into 30 slices to provide enough chroma definition while not creating unnecessary analysis values that might hinder the efficiency of the model. The analysis dataset is stored in a buffer then flattened into a more easily manageable format. The chroma analysis will be used as a feature for prediction in our multi-layer perceptron classifier.

2.4.2 Model training

If the model was not pre-trained, we first need to fit the model. One model could be trained for the whole ensemble since the ensemble should share the same tuning system and we are only seeking for tone class. The user – assumed to be an experienced *gambang* musician - has the tools to train

[4] e.g., pieceNameTake0_section_00-INSTRUMENT.wav

the model themselves. They choose a *ketukan*. The sound stored at that position in the buffer is played for the user to listen to and associate the appropriate note value to it. We select a few (i.e., 5 to 10) instances of each tone and associate it to a number (i.e., 1 to 7; *ding* to *dong cenik*). We obtain from this association a pair comprised of a note value as a label, with an array of 30 chroma analysis values as a feature. When enough associations have been realized, the model is ready to be fitted. The model is trained by sending a fit message to the [fluid.mlpclassifier~] object, a multi-layer perceptron. We used the first *kilitan* section of *Semarandana* as a training set.

2.4.3 Model prediction

When the model is trained, we test it by activating the prediction phase. We select one stroke to be simultaneously played and analyzed through the chroma analysis system. The analysis and prediction processes run in parallel for all five instruments for a selected *ketukan*. The model predicts a label for each analysis and outputs a predicted note value ranging from 1 to 7. If no onset was detected at the selection, no analysis is performed, and we obtain an output value of 0. The model can be saved to disk for reuse at a later time. This is particularly useful if a group regularly uses the same set of instruments and wishes to perform recurrent transcriptions.

2.5 Grid-like transcription

The result of the model's prediction is stored in a [jit.cellblock] grid-like transcription holding information on note values per instrument (column) at each *ketukan* (row). The data can be adjusted manually before exporting to a .csv file [12].

```
1  Pupuh Gangsa A B C D,
2  0 5 0 6 0 5,
3  0 0 5 3 1 0,
4  0 0 1 5 3 5,
5  0 0 3 1 5 0,
6  0 0 0 5 0 5,
7  0 0 5 3 1 0,
8  0 0 1 6 0 1,
9  0 0 5 5 5 0,
10 0 0 0 3 0 6,
11 0 0 3 0 3 0,
12 0 3 3 6 0 1,
13 0 0 1 1 1 0,
14 0 0 3 0 3 5,
15 0 0 0 3 5 0,
16 0 0 1 5 0 6,
17 0 0 3 3 3 0,
18 0 1 0 0 1 1,
19 0 0 1 1 0 0,
20 0 0 0 3 3 3,
21 0 0 3 6 5 1,
22 0 0 1 1 0 0,
23 0 0 3 1 1 3,
24 0 0 0 6 3 1,
25 0 0 1 5 5 0,
26 0 0 1 3 0 3,
27 0 0 3 0 3 0,
28 0 3 3 6 0 3,
29 0 0 1 1 1 0,
30 0 0 3 0 3 3,
31 0 0 0 3 5 0,
32 0 0 1 0 0 1,
33 0 0 3 1 3 0,
34 0 5 0 5 0 5,
35 0 0 5 5 5 0,
36 0 0 1 3 0 5,
37 0 0 3 1 3 0,
38 0 0 0 5 0 5,
39 0 0 5 3 5 0,
40 0 0 1 5 0 6,
41 0 0 5 6 3 0,
42 0 0 0 6 1 1,
43 0 0 1 1 0 0,
44 0 1 0 1 0 5,
45 0 0 3 6 6 0,
46 0 0 1 1 0 6,
47 0 0 3 1 5 0,
48 0 0 0 3 3 1,
49 0 0 1 1 5 0,
50 0 6 0 6 0 6,
```

Figure 3: Grid-like representation obtained by our algorithm of the first 49 *ketukan* of the *kaping dua* section of the piece *Semarandana*, interpreted by Sanggar Seni Naradha Gita. Columns in order from left to right represent: *pupuh*, *gangsa*, *cungklik pemetit* (identified as 'A'), *cungklik penyelat* (as 'B'), *cungklik pemero* (as 'C'), and *cungklik pengede* or *pengenter* (as 'D').

3 Results

What we are presenting is a first version of the algorithm, so the results are necessarily preliminary. We used a 753 *ketukan* long excerpt (second *kilitan*) of the piece *Semarandana* as a testing set. A human compared the original recording to the transcription rendered as audio using sampled instruments. We validated manually whether the algorithm correctly identified each onset and tone class. The transcription obtained a F-measure [10] of 0.967, with a precision of 0.971 and recall of 0.964, which we consider a promising result as a base before refining the system.

4 Conclusion

The algorithm is currently limited to automatic transcription of audible sounds (i.e., *gangsa* and *cungklik*). This creates a relatively accurate transcription of one instance of an interpretation of a piece, making it a useful tool for musicological purposes. However, it should not be considered a valid *gambang* notation for instrumental practice since the musical style relies on improvisation. The tool proposed is a step towards the automation of the transcription process of *pupuh gambang* repertoire. The use of this tool unfortunately requires multichannel recoding equipment and recording technique knowledge, which is not readily available for most users in Bali. A next step will be to conduct multitrack recordings of more *gambang* groups in order to broaden the corpus. This would also provide the opportunity to test further the transcription algorithm. The transcribed repertoire would then be sufficient in size to eventually train deep learning models in the generation of *gambang* music.

Acknowledgments

This research is supported by Observatoire interdisciplinaire de création et de recherche en musique (OICRM). We thank the musicians and members of Naradha Gita gamelan ensemble; I Nyoman Astadi Jaya Pramana, I Made Putra Antara, I Made Agus Tara, Kadek Sumanjaya, I Gede Gili Mahendra Putra, I Gede Hery Sudharma, I Ketut Jambenegara; and their mentor; I Wayan Budiarta; who provided their knowledge and skill, and to Zachary Hejny who contributed with the instruments' audio samples.

References

[1] Adams, J. S. (2021). *Pupuh Gambang: Manuscript, Melody, and Music* [University of British Columbia]. https://doi.org/10.14288/1.0397004

[2] Adams, J. S. (2021, October 26). *Communication with Jonathan Adams on the Acquisition of Data for the Dissertation "Pupuh Gambang: Manuscript, Melody, and Music"* [Personal communication].

[3] Benetos, E., Dixon, S., Duan, Z. and Ewert, S. (2019). Atomatic Music Transcription: An Overview. *IEEE Signal Processing Magazine*, 36(1), 20-30. https://doi.org/10.1109/MSP.2018.2869928

[4] Gomez, E., Haro, M., & Herrera, P. (2009). Music and Geography: Content Description of Musical Audio from Different Parts of the World. *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*.

[5] Sinti, I. W. (2011). *Gambang: Cikal Bakal Karawitan Bali* (Edisi I). TSPBOOKS.

[6] Rembang, I. N. (1984). *Hasil Pendokumentasian Notasi Gending-Gending Lelambatan Klasik Pegongan Daerah Bali* (Departmen Pendidikan Dan Kebudayaan Direktorat Jendral Kebudayaan Proyek Pengembangan Kesenian Bali).

[7] Cycling '74. (2018). *Max* (Version 8) [C, C++ (on JUCE platform); Windows, macOS]. https://cycling74.com/products/max/

[8] Tremblay, P. A., Green, O., Roma, G., Bradbury, J., & Moore, T. (2022). *Fluid Corpus Manipulation* (FluCoMa Max Nightly Release) [C++, SuperCollider, Max, Python, JavaScript; macOS, Windows, Linux]. https://github.com/flucoma

[9] Roma, G., Xambó, A., Green, O., & Tremblay, P. A. (2021). A General Framework for Visualization of Sound Collections in Musical Interfaces. *Applied Sciences*, *11*(24), 11926. https://doi.org/10.3390/app112411926

[10] Ajmera, J., McCowan, I. A., & Bourlard, H. (2004). *Robust Audio Segmentation* [PhD Thesis, École Polytechnique Fédérale de Lausanne]. https://infoscience.epfl.ch/record/83070

[11] Shepard, R. N. (1964). Circularity in Judgments of Relative Pitch. The Journal of the Acoustical Society of America, 36(12), 2346–2353. https://doi.org/10.1121/1.1919362

[12] Murray, A. (2021). *Max CSV / TSV Tools* [Max, JavaScript; Windows, macOS]. https://github.com/adamjmurray/max_csv_tools (Original work published 2010)