RIPPLE: integrated audio visualization for livecoding based on code analysis and machine learning

Hiroki Matsui *
g3119019aa@edu.teu.ac.jp

Keiko Ochi*
ochikk@stf.teu.ac.jp

Yasunari Obuchi*
obuchiysnr@stf.teu.ac.jp

*Tokyo University of Technology

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Copyright remains with the author(s). *ICLC 2020*, February 5-7, 2020, University of Limerick ,Limerick, Ireland

Introduction

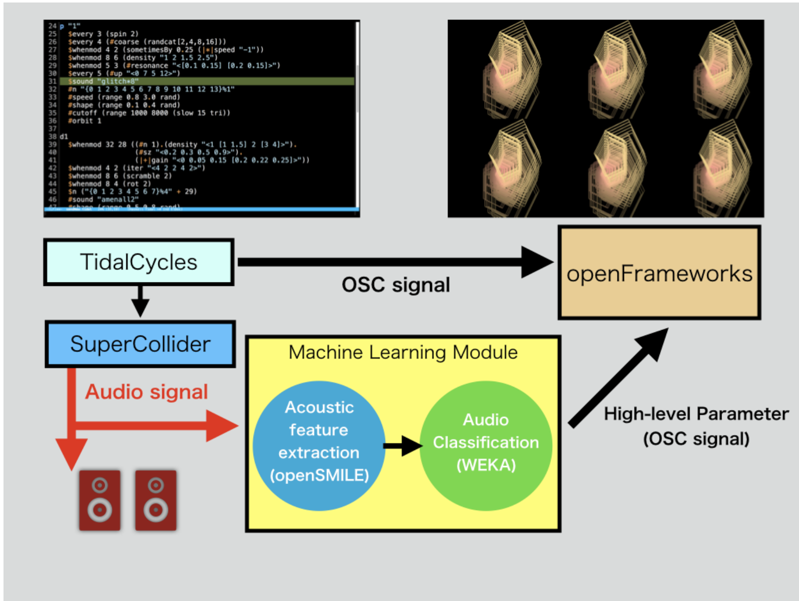

Figure 1: RIPPLE overview

We have created an integrated audio-visual performance system called RIPPLE for livecoding. In this system, in addition to the OSC signal generated by the LC environment, high-level music information calculated by real-time audio analysis by machine learning is used for visualization. The system enables real-time generation of music and visuals by single coder.

RIPPLE

The coder plays TidalCycles. The ML module analyzes the generated audio to derive high level information. Finally, the machine learning output and Tidal-OSC are visualized with openFrameworks.

Machine Learning

Label Category	Label 1 (#slices)	Label 2 (# slices)
Brightness	Bright (643)	Dark (240)
Rhythmic Order	Orderly (771)	Disorderly (112)
Sound Organicity	Organic (372)	Mechanical (511)
Harmonicity	Pitched (617)	Noisy (266)
Note Density	Sparse (413)	Dense (470)

Table 1.1: Labeling results

High level information is calculated in the following flow.

- Record performance audio data in real time
- Extract acoustic features(e.g. energy, spectrum) with openSMILE from segments divided into 5 seconds.
- Estimate the label value from the extracted features using SMOreg with reference to the constructed model.

In our model construction example, Matsui's Tidal performance audio data (20 tracks) were used, and each was divided into 10 seconds to make a total of 884 segments (147 minutes). Next, 5 types of binary labels were assigned to each segment, and a model was constructed with SMOreg. Table 1.1 shows the labeling results.

Visualization

Figure 2 shows examples of visualization. Tidal OSC signals and machine learning output are mapped to the polygonal rotational movement as shown in Table 1.2. High-level information extracted every few seconds is reflected in the elements that determine the overall visual, such as color tone and object placement, and Tidal-OSC brings synchronization between music rhythm and video rhythm.

Discussion

Our future works are:

- Model building takes time and effort, so methods and flows need to be improved.
- In addition to the oF that needs to be implemented, it would be useful to be able to link with existing visual environments and Visual Coder.
- The live machine learning flow is invisible, so we want to show it on the screen.

Source	Parameter	Role in Visualization
TidalCycles	Sound Trigger	Animation Trigger
	Effect Value	Size of Object
	Code length	Number of object vertics and screen divisions
	Brightness	Alpha value for screen fill
Machine Learning	Rhythmic order Organicity	Align objects Hue and line weight
	Harmonicity	Saturation of color
	Note density	Number of Objects

Table 1.2: Sound parameters and their roles in visualization

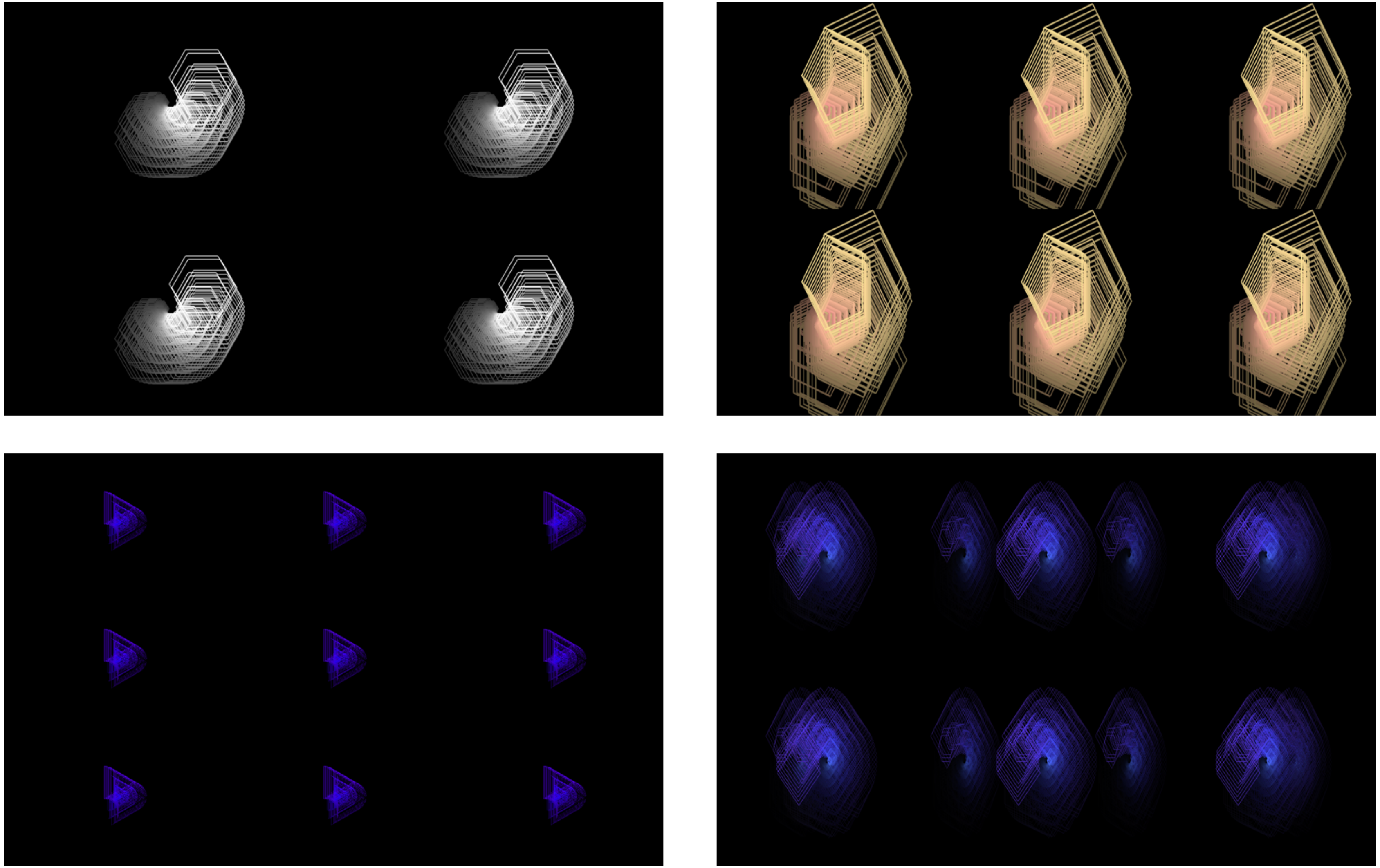

Figure 2: Results of Visualize

References

Cooper, M., Foote, J., Pampalk, E., Tzanetakis, G. (2006). 'Visualization in audio-based music information retrieval', Computer Music Journal, 30(2), 42-62.

Florian, E., Martin, W., and Bjorn, S. (2010) 'Opensmile: the munich versatile and fast open-source audio feature extractor', Proceedings of the 18th ACM international conference on Multimedia, 1459–1462.]

Mark, H., Eibe, F., Geoffrey, H., Bernhard, P., Peter, R., and Ian, H.W. (2009) 'The weka data mining software: an update', ACM SIGKDD explorations newsletter, 11(1), 10–18.

Street, Z., Albornoz, A., Bell, R., John, G., Jack, O., Knotts, S.,McLean, A., Smith, N.C., Tadokoro, A., van der Walt, J.S., Velasco, G. R. (2019) 'Towards Improving Collaboration Between Visualists and Musicians at Algoraves', International Conference on Live Coding.