Real-Time Pitch Estimation for Creative Music Game Interaction

Peter Meier[1,2], Simon Schwär[2], Gerhard Krump[1], Meinard Müller[2]
[1] *Deggendorf Institute of Technology, 94469 Deggendorf, Germany*
[2] *International Audio Laboratories Erlangen, 91058 Erlangen, Germany*
{peter.meier, gerhard.krump}@th-deg.de, {simon.schwaer, meinard.mueller}@audiolabs-erlangen.de

Abstract

Music-based games are an important genre in the gaming community and have become increasingly popular with games like SingStar and Guitar Hero. These types of games are usually based on reactive game mechanics, where the player must hit a certain note at a certain time in order to score points. In this contribution, we present a game prototype that goes beyond purely music-reactive game mechanics and focuses more on the creative aspect of making music in games. In particular, we developed a jump-and-run game that can be controlled with a gaming controller but also uses the player's singing voice to interact with the game world. To this end, we estimate the pitch of a microphone signal in real time and use it as a creative input to the game. This input can be used to control parts of the game world, for instance by singing and adding stair-like elements that allow the player to overcome obstacles and reach the end of a game level. With our game prototype, we demonstrate how game designers can incorporate musical challenges into a well-known game environment while motivating musicians to creatively explore and practice their musical skills.

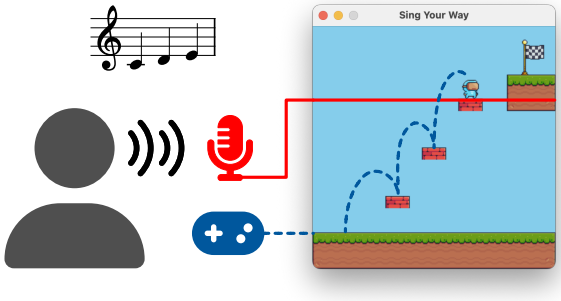

Figure 1: A demo level from our "Sing Your Way" music game prototype. It uses a combination of traditional gaming controller and singing voice input to add note blocks to the game and to reach the end of a level.

Creative Music Game Interaction

This paper is part of a series on "Real-Time Signal Processing Algorithms for Interactive Music Analysis Applications" [1]. In previous work, we presented a music-reactive game with beat tracking, where the game world is generated in real time from ambient music (e.g., radio, live music) of the player [2]. In this article, we consider a different type of music game that uses pitch estimation as a control input. In developing this game, we want to encourage the creative use of melody and singing for players and game designers and concentrate on exploring and teaching real-time aspects of pitch estimation. In particular, we focus on two pitch estimation algorithms, YIN [3] and SWIPE [4], which we adapted from the Python library libf0 [5] and modified to work in real time for our gaming context.

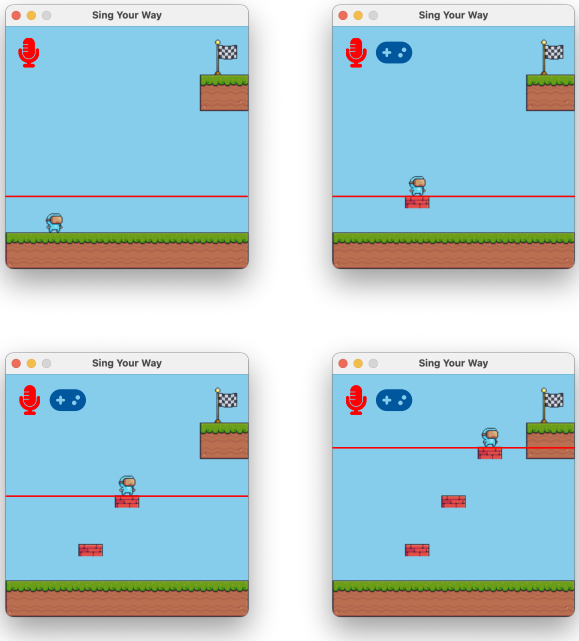

Figure 2: Gameplay sequence showing how to add note blocks and stairs to the game: Sung notes appear as a red line that can be jumped on. This creates a fixed note block on which the player can stand. The pitch of the notes corresponds to the vertical position of the note blocks.

We developed a music game prototype "Sing Your Way" in which players can interact with the game world by using their singing voices. Figure 2 shows what this might look like for a simple use case. In the first step, the player can sing a note. The pitch of this note is displayed as a red line in the game. The higher the note is sung, the higher the red pitch line is displayed. While singing a note, the player can interact with this red pitch line by jumping on it. By doing this, a fixed note block is created in the game for the player to stand on. The same step can be repeated several times at different pitches. As a result, players can add stair-like elements to the environment, using their voices to overcome obstacles, avoid enemies, or reach the end of a level.

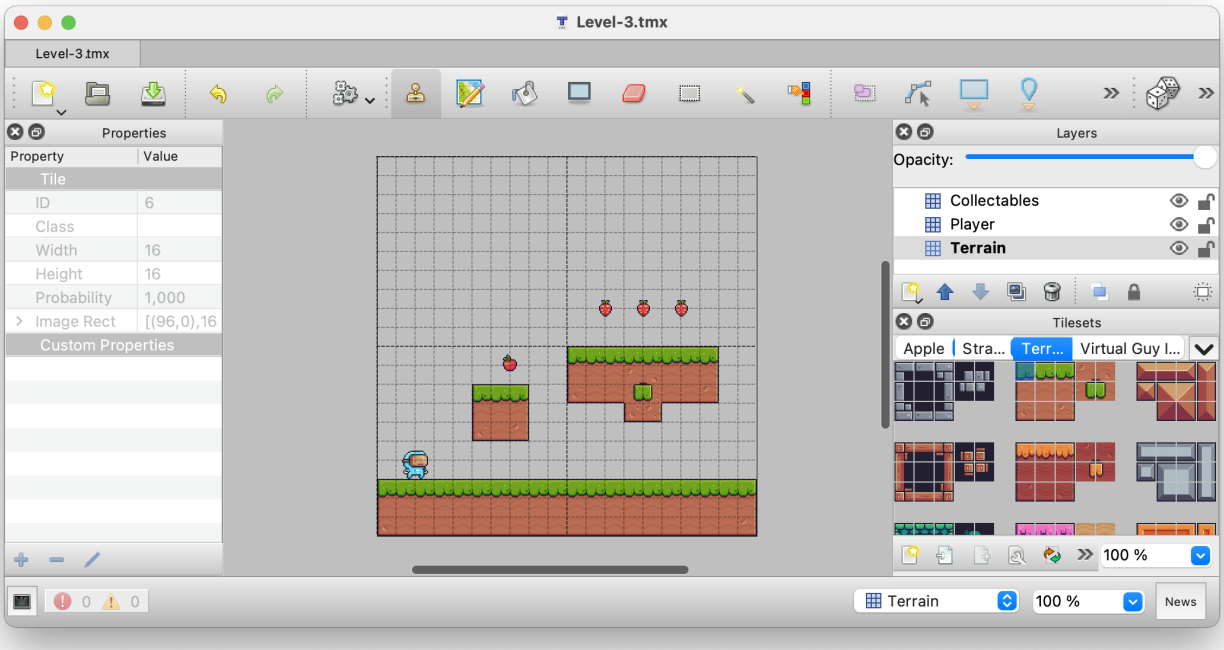

Figure 3: Simple and easy-to-use level design with the free and open source level editor "Tiled." It uses sets of simple graphical elements, called "tilesets", that can be placed on different layers of the map to create a playable level.

A second game mode is shown in Figure 4. By pressing and holding a button on the gaming controller, the player can activate "flight mode". In this mode, the player's vertical position is controlled by the singing voice. The higher the player sings, the higher the player flies in the level, while horizontal movement is still controlled by the gaming controller. This expands the control options of the game and allows the player to reach certain areas of a level that would otherwise be unreachable by simply jumping and running.

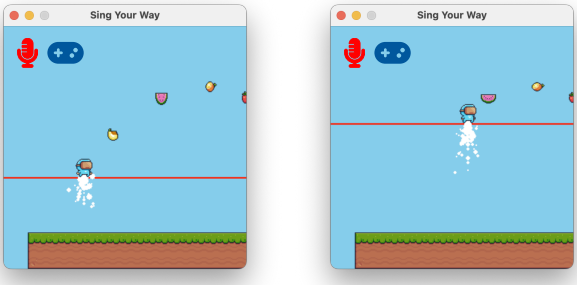

Figure 4: A gameplay sequence showing how to use continuous singing to make the character fly and collect items. In "flight mode", the vertical position of the character corresponds to the pitch of the notes. The horizontal position of the character is still controlled by a gaming controller.

One of the main goals of our game prototype is to create a platform that connects scientists, level designers, and musicians, especially in an educational context. For this reason, we have integrated the ability to design levels using a map editor[1] and a tileset[2], as shown in Figure 3. Firstly, this allows for creative level design without any technical background in programming. Secondly, it gives researchers new and creative ways to test real-time algorithms in the context of a music game. Finally, this platform could also be interesting for musicians to creatively explore and practice their musical skills.

Real-Time Pitch Estimation

In order to use the player's singing voice as an input controller for the game, we need to estimate the pitch of the singing voice, which is a common task in Music Information Retrieval (MIR). The pitch estimation algorithm used for this task has to meet the following requirements to create a smooth and enjoyable gameplay experience. The algorithm must be computed in real time with low latency. In addition, the pitch estimation must be accurate and there must be robust voicing detection that can distinguish between singing and non-singing, as we only want to display a red pitch line when there is actual singing. To find suitable solutions for these requirements, we will first take a closer look at the general challenges of "Real-Time Music Information Retrieval" and then look at two concrete examples of "Pitch Estimation Algorithms" in the following two subsections.

[1] `https://www.mapeditor.org`
[2] `https://pixelfrog-assets.itch.io/pixel-adventure-1`

Real-Time Music Information Retrieval

Music Information Retrieval (MIR) is an area of research that focuses mainly on the analysis of large music datasets [6]. The algorithms used for these analyses are often not optimized for execution time and typically work offline. When it comes to analyzing in real time, there are three major differences compared to analyzing offline, as illustrated by Figure 5.

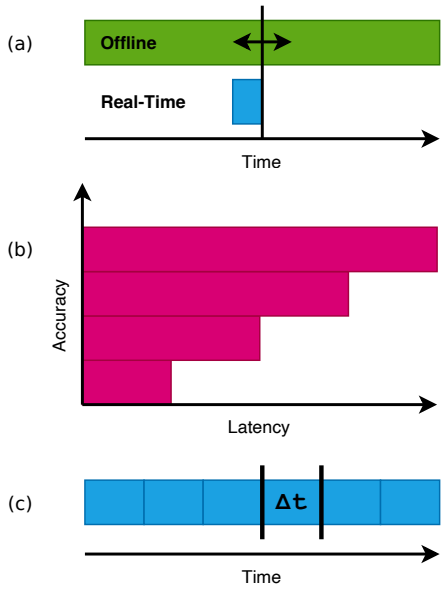

Figure 5: Illustration of three main differences between offline and real-time Music Information Retrieval. (a) Causality of information. (b) Trade-off between accuracy and latency. (c) Signal processing with buffers.

(a) Causality of information. For offline analysis, the complete audio signal is already available and you can move forward or backward in the signal and select analysis windows of any size. For real-time analysis, only past and present information is available, with typically much smaller analysis windows.

(b) Trade-off between accuracy and latency. In general, the larger the analysis window, the more robust the analysis results can be. For instance, for a Fourier transform, a larger analysis window in the time domain would result in more accurate frequency resolution. However, for real-time analysis, this also has a direct impact on latency, as larger analysis windows also mean longer waiting times, especially for causal systems.

(c) Signal processing with buffers. Real-time signal processing is done in buffers. This means that there is only a very short time to analyze a buffer before the next buffer has to be processed. As a result, analysis methods cannot be as complex as desired and must be optimized for execution time.

Pitch Estimation Algorithms

In addition to the general challenges of real-time analysis discussed in the previous section, we now want to take a closer look at the specific challenges of real-time analysis with respect to different pitch estimation algorithms. Pitch is a perceptual property, defined as how "low" or "high" a tonal sound is perceived. When we talk about pitch estimation in this article, we actually mean fundamental frequency estimation (F0 estimation), which is often used synonymously with pitch estimation. The basic idea of F0 estimation is to find the signal's fundamental frequency that correlates with the perceived pitch for each time position. The input to the algorithm is the waveform of the signal. The output of the algorithm is the frequency estimate of the input signal and an indication of how confident the algorithm is that the signal is tonal and periodic, called the "confidence value." This confidence value is often used in combination with a threshold to provide a form of voicing detection, for example, to distinguish singing from non-singing signals.

Two methods that compute the values on a frame-by-frame basis, and are therefore in principle suitable for real-time applications, are shown in Figure 6 and will be discussed in the following section.

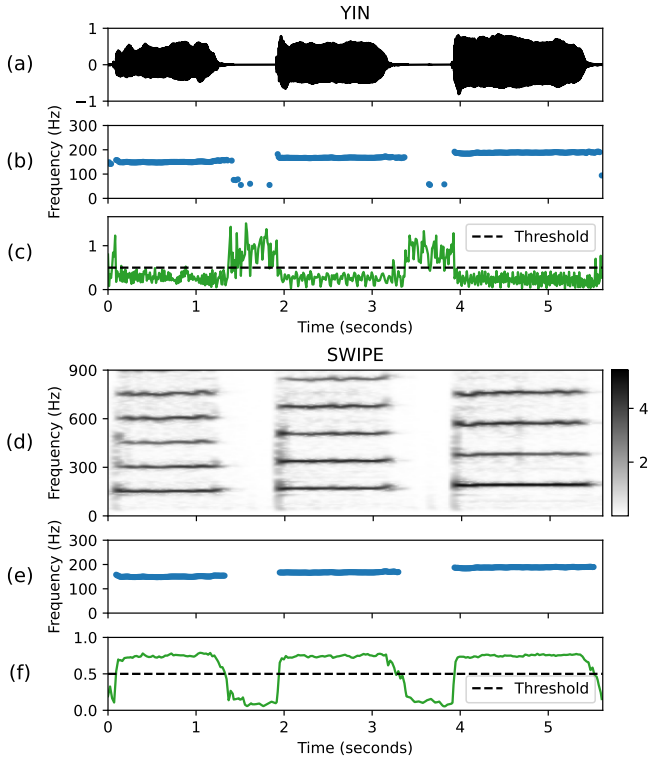

Figure 6: An overview of two pitch estimation algorithms used for our music game prototype. YIN: (a) Waveform input. (b) Pitch estimation trajectory. (c) Aperiodicity as confidence value for voicing detection. SWIPE: (d) Spectrogram input. (e) Pitch estimation trajectory. (f) Pitch strength as confidence value for voicing detection.

YIN is based on an autocorrelation method and is calculated entirely in the time domain, using the signal's waveform as input. The algorithm provides accurate pitch estimates for signals with singing, but also incorrectly detects pitches for non-singing parts of the signal. YIN calculates the so-called "aperiodicity" of the input signal as a confidence value. The smaller the aperiodicity, the

more likely it is that the signal is periodic and tonal, such as a singing signal. A major advantage of YIN with respect to real-time applications is that the algorithm is relatively simple and can be implemented in an efficient way with low latency. However, the voicing detection is not very robust, so there are often outliers in the pitch detection. Nevertheless, YIN can be extended with post-processing steps [7], which we would like to investigate more closely for real-time applications in the future.

SWIPE is based on a spectral correlation method and uses the spectrogram of the signal as input. The algorithm calculates accurate pitch estimates by spectrally correlating the signal with a set of frequency kernels and finding the kernel with the highest correlation score. This correlation score, also called "pitch strength", is used as a confidence value for the algorithm and allows for robust voicing detection. However, the algorithm is quite complex, as it requires the calculation of multiple spectral correlation functions and multiple spectra with different window sizes for each frame. Our initial investigations showed that SWIPE provides lower confidence values when only causal information is available, which affects voicing detection for real-time applications. This is a topic we will explore in more detail in the future.

Conclusion and Outlook

With our music game prototype, we provide a platform that uses real-time pitch estimation as a creative input to an interactive game. Our general aim is to bring together researchers, level designers, and musicians to creatively explore algorithms, design interesting levels and practice musical skills. For future research, we would like to further analyze and evaluate the real-time algorithms presented in this paper with common MIR metrics [8] on publicly available datasets [9], and look at other pitch estimation algorithms based on both classical engineering [7], [10] and more recent machine learning techniques [11], [12].

Acknowledgements

This work was supported by the BayWISS Joint Academic Partnership "Digitalisation." The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen IIS.

References

[1] P. Meier, G. Krump, and M. Müller, "A real-time beat tracking system based on predominant local pulse information," in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021.

[2] P. Meier, S. Schwär, S. Rosenzweig, and M. Müller, "Real-Time MIR Algorithms for Music-Reactive Game World Generation," in *Mensch und Computer 2022 - Workshopband*, Bonn, 2022. DOI: 10.18420/muc2022-mci-ws03-225.

[3] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music.," *Journal of the Acoustical Society of America (JASA)*, vol. 111, no. 4, pp. 1917–1930, 2002.

[4] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.

[5] S. Rosenzweig, S. Schwär, and M. Müller, "Libf0: A Python Library for Fundamental Frequency Estimation," in *Late Breaking Demos of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.

[6] D. Stefani and L. Turchet, "On the challenges of embedded real-time music information retrieval," en, *Proceedings of the 25-th Int. Conf. on Digital Audio Effects (DAFx20in22)*, vol. 3, pp. 177–184, Sep. 2022.

[7] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 659–663.

[8] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "MIR_EVAL: A transparent implementation of common MIR metrics," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 367–372.

[9] R. M. Bittner, M. Fuentes, D. Rubinstein, A. Jansson, K. Choi, and T. Kell, "mirdata: Software for reproducible usage of datasets," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 99–106.

[10] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012. DOI: 10.1109/TASL.2012.2188515.

[11] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 161–165. DOI: 10.1109/ICASSP.2018.8461329.

[12] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirovic, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1118–1128, 2020.