Procedural Rhythm Game Generation in Virtual Reality

Devin Perry
University of South Florida, USA

Thomas Bivins
University of South Florida, USA

Bianca Dehaan
University of South Florida, USA

Wanwan Li*
University of South Florida, USA

ABSTRACT

Rhythm games and Virtual Reality (VR) often find themselves coming together through the help of various applications designed with the intent to deliver a unique gameplay experience to individuals who are seeking an immersive virtual environment where there is background music. Popular rhythm games such as Beat Saber make the gameplay experience unique by allowing users to move and slice multicolored tiles to the beat of developer-made "beat maps." Using music synthesis software, we have designed an application utilizing automatic beat mapping techniques to automatically synthesize rhythm games like Beat Saber in virtual reality to make the game itself more immersive and enjoyable when allowing the user to play any song that they desire, without the need to make the beat maps manually. Through a series of experiment results and user studies, we show that our proposed approach for rhythm game synthesis is immersive and enjoyable when the game objects are automatically synchronized with the background music's rhythm.

CCS CONCEPTS

• **Computing methodologies**; • **Computer graphics**; • **Graphics systems and interfaces**; • **Virtual reality**;

KEYWORDS

rhythm game, beat detection, entertainment, game content synthesis

ACM Reference Format:

Devin Perry, Thomas Bivins, Bianca Dehaan, and Wanwan Li. 2023. Procedural Rhythm Game Generation in Virtual Reality. In *2023 The 6th International Conference on Image and Graphics Processing (ICIGP 2023), January 06–08, 2023, Chongqing, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3582649.3582664

1 INTRODUCTION

In recent years, VR has dramatically shifted from being exclusively a laboratory tool to the latest and greatest household gaming and exercise medium. This steady evolution was made possible with the creation of gaming and exercise applications such as Beat Saber, which modernize the exercise experience to be more than simply a checkbox on a daily to-do list. Interactive VR applications make the exercise experience exciting by creating an atmosphere that cannot be achieved anywhere other than a VR environment. While interactive and physical rhythm games such as Beat Saber may be revolutionary in making exercise an enjoyable experience for all users, it still lacks one key piece that would make the experience even more immersive. The number of songs that are offered within the game for the user to play is extremely limited, as all beat maps are created manually.

It is proven that music positively affects an individual's performance during an exercise session stork et al. [14] has done. Therefore, to reach maximum levels of immersion and efficiency, users should be able to input any song that they desire to exercise to. Not only would the user feel more immersed in the VR environment, but the user would achieve a more calorie-burning exercise while moving to a song that they personally chose and enjoy. [5] This can be achieved using music detection and audio synthesis software to analyze audio tracks to generate automatic beat maps. Games like Beat Saber and other commercial music-themed VR titles often ship with a limited library, due to contractual limits and licensing concerns in implementing tracks on a paid title. In the case of many of these games, however, many users have managed to implement their own tracks into the game through modifications like BMBF, a user-created patch for Beat Saber that "allows for custom songs and modifications" within Beat Saber. These modifications rely on users to "map" beats manually to a track and make their own layouts, limiting the library to tracks that users curate.

Implementation of a procedural way to detect the beats and map tiles to these beats would allow users to generate tracks without manual placement, opening the library to any track that can be procedurally mapped. Research into this field has been active, with notable papers such as Al-Hussaini et al.'s research into "Predictive Real-Time Beat Tracking from Music for Embedded Application" [1] in 2018, demonstrating results that "can satisfactorily estimate beat positions from a music signal in real-time," and Benetos et al.'s "Automatic Music Transcription: An Overview" in 2019, with a focus on "converting music signals into some form of music notation" via "signal processing and artificial intelligence." [2] While both of these papers cover a breadth of ways to transcribe and map audio tracks, a focus on utilizing this research in the field of VR has not been extensively covered. Because these tracks have the benefit of being generated before the user plays, the tradeoff between accuracy and speed lends to an accuracy-focused approach to signal analysis.

2 RELATED WORKS

In 2015, Stork et al. [14] proposed a novel approach to the introduction of music-enhanced physical performance and its impacts

*Wanwan Li is the corresponding author. Devin Perry, Thomas Bivins, Bianca Dehaan are the equal Contributors

ICIGP 2023, January 06–08, 2023, Chongqing, China

ACM ISBN 978-1-4503-9857-2/23/01…$15.00
https://doi.org/10.1145/3582649.3582664

on aversion to sprint interval training which inspired our work toward how we could implement beat detection automatically. In 2021, Shimizu et al. [13] presented Swaying Locomotion, a locomotion system based in VR that supported users moving in a stationary VR space, that inspired the team to utilize full body motion while capturing positions as opposed to traditional joystick-based approaches. In 2021, Nagasaka et al. [11] proposed advanced visualizations of Deep Learning (DL) models that heavily impacted our work in implementing the synthesis of Audio inputs. In the same conference (VRST December 2021), Blaga et al. [4] presented categorization models for VR interaction, foundational for understanding the interactions between user sabers and beat tiles for this research. In 2021, Dalsgaard et al. [6] released research that sought to find utilize natural features to implement pointing features for distant objects in VR, which our team used to influence how the motion tracking and directional selection for beat tile interaction would happen. The same year, Bhardwaj et. al [23] developed key visualization tools to study the effects of mid-air haptics, which is reflected in our research focusing on using beat detection to enact actions resulting in similar mid-air haptics for the user. Bimberg et al. [3] released in 2021 a study focused on maneuvering in immersive virtual environments, which inspired our focus lock for maintaining direction in the user for the actual gameplay. Another study, one that focused on the striking judgment in VR, done by Yanase et al. [16] would contribute to the implementation of bat and saber striking in the VR environment In 2020, Esmaeili et al. [7] research on Scaled Hand Interactions in the VR space, specifically the focus on "natural physical hand interactions," inspired our team in the best way to implement the VR interactions with the blocks in the system. Romero et al. [24], 2018 discussed foveated rendering capabilities in VR to reduce cognitive overload, which we used in order to reduce user overload while in the Unity scene. In an ACM Research-Article by Hunt et al. [9] in 2018, a visual hierarchy is developed with real-time performance in mind, specifically focusing on lower memory bandwidth, something directly inspiring our work in streamlining performance mid-play. In 2018, Hussaini et al. [1] research on real-time beat tracking was crucial for implementing Fourier-transformed beat mapping. Schwelling et al., [12] in the same year, published a paper focusing on automated modeling and visualization of audio in a VR space that directly impacted the team in how we approached the translation of beat mapping to tile placement in the play space. Besides, other works such as VR for lines production's simulation [18], the SPIDAR-if [19], the electro-tactile feedback system for VR [20], analysis of the growth perspectives [21], and the design of VR prototyping system [22] are also relevant to our research work.

A similar spatial concept was researched a year later by Yoo et al. in their 2019 paper "Spatially Accurate Generative Music with AR Drawing," [17] necessary for our work in focusing the player within a music-generated play space like Beat Saber. Much of this research and development would not be possible without the work of many researchers in the Beat Saber and rhythm game genre, such as Sumitomo et al.'s research on the range of motion within VR music games in 2020 [15], work that focused on the range of motion necessary for games such as Beat Saber. In addition, Kim et al.'s 2021 research on deep-learning-based music analysis through visual methods like object detection [10], an alternative to audio-based beatmap extraction inspired our work in audio-based mapping. Estolas et al.'s 2021 research [8] into automatic beat map generation and genre detection in their own health and reflex-focused titles gave crucial insight into applications of signal analysis for our team's work.

3 TECHNICAL APPROACH

The audio spectrum is the range of frequencies that humans can hear, ranging from 20 Hz to 20,000 Hz. This range can be broken down into seven bands of frequency ranges, splitting between sub-bass, bass, low-midrange, midrange, upper-midrange, presence, and brilliance.

Sub-bass, in the range of 20-60 Hz, is heavily impacted by the Fletcher-Munson curves and results in a bass that is hard to pick up on but can usually be felt. Bass, within the range of 60-250 Hz, hosts the main components of a track's rhythm. Low-midrange typically contains the lowest order harmonics at ranges of 250 to 500 Hz. Midrange stages the prominent instruments and features of a track in the 500-2000 Hz band. Upper, or higher-midrange, sits around 2000-4000 Hz. Presence, containing higher harmonics for instruments such as a violin, is in the 4000-6000 Hz range. Brilliance, the highest frequency band, contains high-pitch elements like whistles and percussion, which sit anywhere from 6000-20000 Hz.

Often, many audio tracks contain multiple (if not all) frequency bands, and inspecting a vibration signal will give the appearance of a distorted sinusoidal wave or a wave impacted by multiple bands resulting in a distorted sine wave. To break a vibrational signal into readable frequency bands, our program utilizes Fourier Transformations, used to break a signal into multiple frequency sine waves.

Frequencies are detected in the audio and split between sub-bass, bass, low-midrange, midrange, upper-midrange, presence, and brilliance. This is done via Fourier Transformations to convert sinusoidal signals into fundamental frequencies. While this requires a finite number of terms, we can consider our track to be within a window function W(t). Given our Discrete Fourier Transformation function.

$$F_n = \sum_{n=0}^{N-1} \left(f_n e^{-2\pi i k n / N} \right) \qquad \text{K p} = \ 0, 1, \ldots.. \text{N} - 1$$

Taking this function and implementing directly (resulting in a computation time of O(Nˆ2)) gives us an output of Amplitude and Phase offset. We can take this Amplitude and plot it along the frequency of the track to obtain the relative frequencies of a given track and break this into the aforementioned frequency bands.

As shown in Figure 1, in our presented program, the Python pyAudio library is able to detect within the given ranges what band each note corresponds to. To accomplish this, the program pulls the audio in via Wavio, to read and write to WAV files via numpy arrays. () This audio is passed into numpy's Fourier Transformations in a reduced sample size to extrapolate the main tempo of the track and split the frequencies into workable bands. This is then ordered by frame in the track via the wave library's getframerate() functions, in order to cut each frame into sortable frequency bands. The max frequencies of each of these bands are set to be the limit for each frequency. As each beat is analysed, if a band exceeds a certain limit

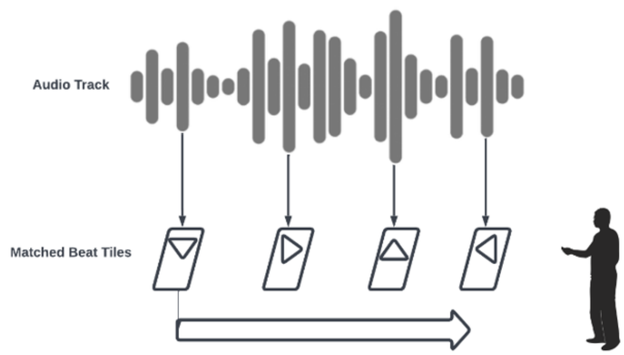

Figure 2: This figure depicts a basic functionality of the beat detecting system; beats are recognized and provide cues as to where beat tiles are to be placed and delivered to the user.

Figure 1: The figure depicts the basic functionality of the beat-detecting system; beats are recognized and provide cues as to where beat tiles are to be placed and delivered to the user.

(currently set to 90% of the maximum frequency), it is logged as a beat. If it dips below a certain frequency (currently set below 30% of the maximum frequency), it is logged as not a beat. This logs all of the tracks as a beat for each frequency band recorded.

Audio Synthesis. The backend of the design utilizes the Python audio synthesis library Librosa, which takes audio input in the form of an mp3 file and can return different information about the file. The beat of the song can be estimated by the software by obtaining the sampling rate and a NumPy array of the audio file along with the total duration of the song. This allows for the best possible estimate for the overall beats. One issue that was found was in the case of beat or rhythm switches in a song, the result would be thrown off and only reflect the beat averages of the first section of the song. This is due to the sampling rate being different in the latter portion of the song, while the duration remains constant. These "beats" are represented as timestamps in seconds that are stored in a txt file. The txt file is then sent to the front to begin the beat tile generation phase.

Beat Tile Output. The front end of the design utilizes the Unity game engine that assists in the generation of the environment, the UI, and the virtual tools that the user will use to interact with the environment. An object in Unity is created whose operation is dictated by a C Sharp script that parses the resultant txt file from the Python backend. This object generates a beat tile that moves toward the user every time the current timestamp of the playing song matches with the current timestamp in the txt file.

This is done via taking in the audio synthesis text file outputs, listing these timestamps in seconds, and adding in necessary delay onto the timestamps to account for tile travel time. The beat tiles come towards the user and are intended to be hit as the beat occurs, so these tiles need to account for travel time with a delay added to these timestamps for generating the beat object in the scene.

On registering these timestamps, the generated timestamps generate blocks in a randomly distributed direction and placement within a grid to be sent to the player. These are registered in the form of "PlacementDirection," so a block in the southwest corner facing upwards would be registered as a "USW" block, which is then parsed through a switch case to instantiate these blocks for the user in real-time. These blocks approach the player with a vector-locked camera, transforming along the axis into the range of the user's hands within the beat time provided by the timestamp.

These RigidBody tiles use Box Collider triggers to interact with the users' controllers, adding to a "score" on the screen on collision and removing the element. This allows multiple beats to be present on the screen but allow for the user to focus on a few at a time with timestamp offsets. One current limitation is the beats are restricted to single input-per-beat actions, and multi-strike beats are currently not implemented. However, these beats can be implemented in a few researched approaches, including random displacement of tiles allowing for pairs, triplets, and quadruplet actions, pre-generated "chunks" of beats that could be generated alongside the single beats, and utilizing frequency bands beats to generate different patterns.

4 USER STUDY

We tested our approach on the three different types of Procedural Rhythm Game Generation in virtual reality using three different pieces of music. The three different game types are shown in Figure 2. Beat Saber (Galaxy scene using Bad Romance by Lady Gaga), Figure 3. Fruit Ninja (Room scene using Despacito by Luis Fonsi and Daddy Yankee), and Figure 4. Baseball Field (Stadium scene using Shape of You by Ed Sheeran). The video screen recording of the experiment result is shown at this link: https://youtu.be/zZivD_G_s1A This was done using an Oculus Quest 2, connected via Link Cable to a desktop running the Unity scenes. For each figure, the left shows the scene in Unity Editor, the middle shows the scenes in SteamVR, and the right shows the user. The user was given a headset, the Oculus Quest 2 headset, and corresponding controllers as the experimenter started up the experience scene by scene. The user would then go through all three scenes in sequence before moving on to taking a user study. To conduct a user study, we focused on evaluating user enjoyment and analysis of beat accuracy from the user's perspective. Users were given a demo of each of the scenes to run through and then were provided a survey and a brief interview to expand on any possible notes given by the user. The questions focused on four core focus areas for our research: if the experience was immersive, if the user was comfortable, if the user had fun/found enjoyment, and how well the user believed the beats matched up with the attached song. This feedback was retrieved using a Google form given to the user immediately post-experience, and the interview was done if any suggestions or questions from the users were found during the questionnaire.

5 EXPERIMENT RESULT

Figure 5 shows the user evaluation. After collecting user scores and testing the three VR scenes, our team obtained data both in regards to the user experience and beat matching accuracy, as well as distinguishing dierences between scenes, and what worked versus what needs work. The user study feedback was in response

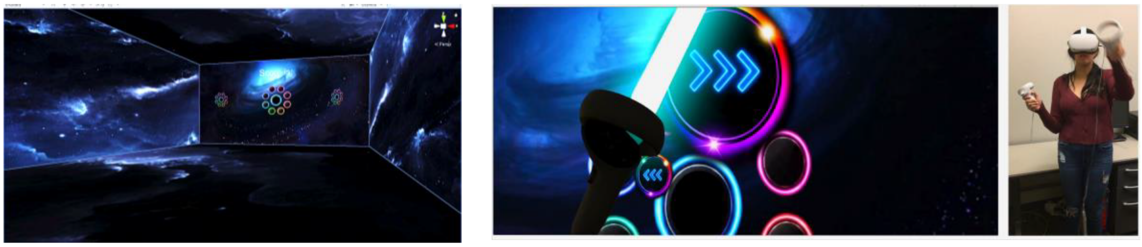

Figure 2: Beat Saber – Galaxy Scene.

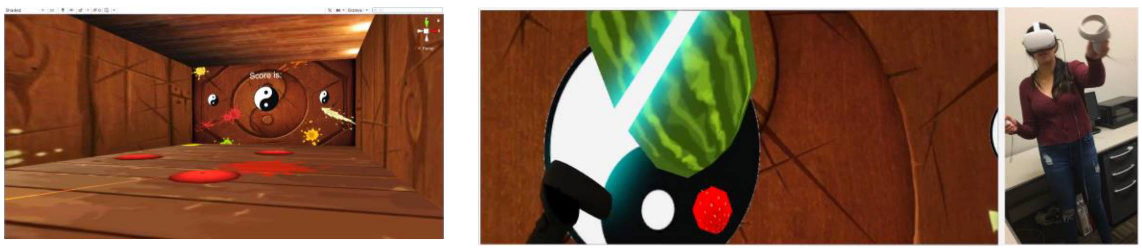

Figure 3: Fruit Ninja – Room Scene.

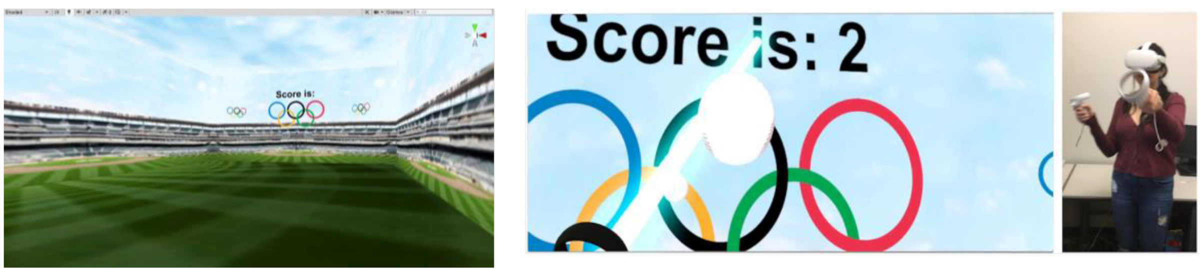

Figure 4: Baseball Field – Stadium Scene.

to comfort (scaled between 1-5), with a mean of 3.2, accuracy of beat matching (scaled between 1-5), with a mean of 4.3, immersion of experience (scaled between 1-5), with a mean of 2.9, and how fun the experience was for the user (scaled between 1-5), with a mean of 4.1. While the comfort and immersion sat around 3, beat matching and enjoyment levels were above the median value, indicating that while the comfort level and immersion were low, the core beat matching algorithm and enjoyment performed well among users.

Users were also polled about their prior experience in VR Rhythm games and how our experience compared to current established products, and if they could see beat matching working well in those other games. The performance of users in the three scenes also depended on prior user usage in the Virtual Reality scene, with users that reported having experience in rhythm-related VR titles scoring ~32% higher than users that reported little to no experience in rhythm titles. Utilization and adaptation of each scene varied, both in how the music fits into the program as well as how the Unity scene would work with the SteamVR integration. The baseline Beat Saber theme performed the best across the board among users (averaging .5 points higher than either of the other scenes), stating that it was the most fun and immersive of all three scenes. Much of

Figure 5: User Evaluation

this is also impacted by the notes themselves, with the Beat Saber scene utilizing easy-to-read directional note blocks, as opposed to the less clear fruit and baseball note blocks in the latter two scenes.

6 CONCLUSION

The team's exploration and adaptation of beat detection in rhythm games proved to be a beneficial tool for utilization in Unity, however, testing in actual third-party games such as Beat Saber was not tested in this experiment. The detection program is able to map these beats to a registerable timestamp, so even manual mapping to this third-party software is feasible. In addition, application to music with variable BPM and shifting tones such as experimental music leaves much to be desired in terms of accurate mapping, as the track is extrapolated from samples taken in by the program, something the team wishes to research further.

REFERENCES

[1] Irfan Al-Hussaini, Ahmed Imtiaz Humayun, Samiul Alam, Shariful Islam Foysal, Abdullah Al Masud, Arafat Mahmud, Rakibul Islam Chowdhury, Nabil Ibtehaz, Sums Uz Zaman, Rakib Hyder, *et al.* 2018. Predictive real-time beat tracking from music for embedded application. In 2018 IEEE Conference on multimedia information processing and retrieval (MIPR). IEEE, 297–300.

[2] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. 2019. Automatic Music Transcription: An Overview. IEEE Signal Processing Magazine 36, 1 (2019), 20–30. https://doi.org/10.1109/MSP.2018.2869928

[3] Pauline Bimberg, Tim Weissker, Alexander Kulik, and Bernd Froehlich. 2021. Virtual Rotations for Maneuvering in Immersive Virtual Environments. In Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology (Osaka, Japan) (VRST '21). Association for Computing Machinery, New York, NY, USA, Article 21, 10 pages. https://doi.org/10.1145/3489849.3489893

[4] Andreea Dalia Blaga, Maite Frutos-Pascual, Chris Creed, and Ian Williams. 2021. Virtual Object Categorisation Methods: Towards a Richer Understanding of Object Grasping for Virtual Reality. In Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology (Osaka, Japan) (VRST '21). Association for Computing Machinery, New York, NY, USA, Article 43, 5 pages. https://doi.org/10.1145/3489849.3489875

[5] Fikana Mahardika Cantri, Zaqiatud Darojah, and Endah Suryawati Ningrum. 2019. Cumulative Scores Based for Real-Time Music Beat Detection System. In 2019 International Electronics Symposium (IES). IEEE, 293–298.

[6] Tor-Salve Dalsgaard, Jarrod Knibbe, and Joanna Bergström. 2021. Modeling Pointing for 3D Target Selection in VR (VRST '21). Association for Computing Machinery, New York, NY, USA, Article 42, 10 pages. https://doi.org/10.1145/3489849.3489853

[7] Shaghayegh Esmaeili, Brett Benda, and Eric D. Ragan. 2020. Detection of Scaled Hand Interactions in Virtual Reality: The Effects of Motion Direction and Task Complexity. In 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). 453–462. https://doi.org/10.1109/VR46266.2020.00066

[8] Elijah Alixtair L. Estolas, Agatha Faith V. Malimban, Jeremy T. Nicasio, Jyra S. Rivera, May Florence D. San Pablo, and Toru L. Takahashi. 2020. Automatic Beatmap Generating Rhythm Game Using Music Information Retrieval with Machine Learning for Genre Detection. In 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM). 1–6. https://doi.org/10.1109/HNICEM51456.2020

[9] Warren Hunt, Michael Mara, and Alex Nankervis. 2018. Hierarchical Visibility for Virtual Reality. Proc. ACM Comput. Graph. Interact. Tech. 1, 1, Article 8 (jul 2018), 18 pages. https://doi.org/10.1145/3203191

[10] Yeonghun Kim and Sunghee Choi. 2021. Vision-based beatmap extraction in rhythm game toward platform-aware note generation. In 2021 IEEE Conference on Games (CoG). 1–5. https://doi.org/10.1109/CoG52621.2021.9619108

[11] Hikaru Nagasaka and Motoya Izuhara. 2021. Interactive Visualization of Deep Learning Models in an Immersive Environment. In Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology (Osaka, Japan) (VRST '21). Association for Computing Machinery, New York, NY, USA, Article 82, 3 pages. https://doi.org/10.1145/3489849.3489956

[12] Elijah Schwelling and Kyungjin Yoo. 2018. Automatic 3D modeling of artwork and visualizing audio in an augmented reality environment. In Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology. 1–2.

[13] Masahiro Shimizu and Tatsuo Nakajima. 2021. Swaying Locomotion: A VR-Based Locomotion System through Head Movements (VRST '21). Association for Computing Machinery, New York, NY, USA, Article 97, 2 pages. https://doi.org/10.1145/3489849.3489897

[14] Matthew J Stork, Matthew Y Kwan, Martin J Gibala, and KA Martin Ginis. 2015. Music enhances performance and perceived enjoyment of sprint interval exercise. Med Sci Sports Exerc 47, 5 (2015), 1052–1060.

[15] Souma Sumitomo and Yuta Sugiura. 2020. VR Music Game Considering Range of Arm Motion. In 2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech). IEEE, 59–62.

[16] Kentarou Yanase, Shunji Muto, Kyohei Masuko, Tomoyuki Nagami, and Takashi Ijiri. 2021. A System for Practicing Ball/Strike Judgment in VR Environment. In Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology (Osaka, Japan) (VRST '21). Association for Computing Machinery, New York, NY, USA, Article 51, 2 pages. https://doi.org/10.1145/3489849.3489931

[17] Kyungjin Yoo and Eli Schwelling. 2019. Spatially Accurate Generative Music with AR Drawing. In 25th ACM Symposium on Virtual Reality Software and Technology. 1–2.

[18] Mohamed-Amine Abidi, Barbara Lyonnet, Pierre Chevaillier, and Rosario Toscano, "Contribution of Virtual Reality for Lines Production's Simulation in a Lean Manufacturing Environment," International Journal of Computer Theory and Engineering vol. 8, no. 3, pp. 182-189, 2016.

[19] Wei-Yu Chen, Makoto Sato, Yen-Ming Chu, and Teruki Honma, "SPIDAR-if: A Joint User Interface Approach for Hapticised Web Content and Augment Reality Enhancement," International Journal of Computer Theory and Engineering vol. 8, no. 6, pp. 444-449, 2016.

[20] D. S. Pamungkas and K. Ward, "Electro-Tactile Feedback System to Enhance Virtual Reality Experience," International Journal of Computer Theory and Engineering vol. 8, no. 6, pp. 465-470, 2016.

[21] Georgi P. Dimitrov, Galina Panayotova, Rosen P. Toshev, and Iva Kostadinova , "Analysis of the Growth Perspectives of the Early Stages of Technological Innovation in the Case of the Augmented and Virtual Reality Application," International Journal of Computer Theory and Engineering vol. 9, no. 6, pp. 443-446, 2017.

[22] G. Ko, S. Ryu, S. Nam, J. Lee, and K. Suh, "Design of Virtual Reality Prototyping System and Hand-Held Haptic Controller," International Journal of Computer Theory and Engineering vol. 11, no. 4, pp. 72-75, 2019.

[23] Bhardwaj, Ayush, Junghoon Chae, Richard Huynh Noeske, and Jin Ryong Kim. "TangibleData: Interactive Data Visualization with Mid-Air Haptics." In Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology, pp. 1-11. 2021.

[24] Romero-Rondón, Miguel Fabian, Lucile Sassatelli, Frédéric Precioso, and Ramon Aparicio-Pardo. "Foveated streaming of virtual reality videos." In Proceedings of the 9th ACM Multimedia Systems Conference, pp. 494-497. 2018.