JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 26, No. 1, pp. 61-68, Jan. 2025

Check for updates

멀티미디어 공연을 위한 동작 인식 시스템

이 관 규[1] · 김 준[2*]
[1]동국대학교 영상대학원 멀티미디어학과 강사
[2]동국대학교 영상대학원 멀티미디어학과 교수

An Interactive System Using Gesture Recognition for Multimedia Performance

Gwangyu Lee[1] · Jun Kim[2*]

[1]Lecturer, MARTE Lab. Dept. of Multimedia, Dongguk University, Seoul 04620, Korea
[2]Professor, MARTE Lab. Dept. of Multimedia, Dongguk University, Seoul 04620, Korea

[요 약]

본 연구는 머신러닝을 활용하여 무용수의 동작을 분류하고, 이를 음악과 영상, 무용이 결합된 멀티미디어 공연에 사용할 수 있는 시스템 디자인에 초점을 맞추고 있다. 이를 위해 아이폰과 CoreML을 사용하였으며, 제스처를 분류하는 전용 앱을 디자인하여 제스처의 일치율을 네트워크를 통해 실시간으로 전송한다. 전송된 데이터는 음악과 영상을 제어하는 데 사용되었으며 이는 멀티미디어 공연에 사용되었다. 본 연구를 통해 접근한 방식은 멀티미디어 콘텐츠를 위한 시스템을 간소화하는 것을 목표로 하며 이는 예술과 기술의 융합을 향상시킨다. 이러한 예술과 기술의 융합은 예술가들이 멀티미디어 공연을 보다 쉽게 제작할 수 있도록 도와주며 본 논문에서 연구된 기술을 활용한 창의적인 멀티미디어 공연을 기대할 수 있다.

[Abstract]

This study focused on developing an interactive system that utilizes machine learning to classify gestures, thereby integrating them into multimedia performances incorporating music, visuals, and dance. The researchers used an iPhone and CoreML in conjunction with a dedicated app designed to classify gestures and communicated the detected gestures and their corresponding levels through a network. The transmitted data are then utilized to control the music and visuals displayed on a computer as part of the interactive multimedia performance. By employing this innovative approach, the study aims to streamline the production of immersive and engaging performances, ultimately enhancing the overall experience for both performers and the audience. This integration of technology and art has the potential to revolutionize the way interactive multimedia performances are created and experienced.

색인어 : 동작 인식, 머신러닝, 음악, 영상, 무용
Keyword : Gesture Recognition, Machine Learning, Music, Visuals, Dance

http://dx.doi.org/10.9728/dcs.2025.26.1.61

Received 30 October 2024; **Revised** 02 December 2024
Accepted 06 December 2024

***Corresponding Author; Jun Kim**

Tel: +82-2-2260-3264
E-mail: music@dongguk.edu

Ⅰ. Introduction

Music, visuals, and dance are different art forms, although they are inseparable. Most music performances incorporate visuals and dance, making it rare to find a music performance devoid of these elements. Traditionally, in dance performances, dancers perform live to music played back from fixed mediums such as CDs or hard drives. In this setup, dancers take their cues from a predetermined, fixed sound world. However, what if dancers could not only follow but also influence, control, or even "play" the music as they danced? What if they could "dance the music" rather than "dance to the music"?[1] Such interactive systems have been attempted in the past[1]-[11]. Examples include changing visuals based on the music's amplitude or the dancer's movement and altering music in response to the dancer's gestures[12],[13].

This study aims to design an interactive system for multimedia performances using gesture recognition, addressing some limitations of prior approaches. For instance, in traditional systems, changes to visuals and music were often triggered by simple parameters, such as the height of a raised hand. While this allowed for basic interaction, it could not distinguish between similar gestures, such as raising a hand sideways versus forward, as both results in an increase in height. To address such ambiguities, an operator often needed to manually intervene to adjust the system's response, which limited the dynamic adaptability of the performance. To address these challenges, this research leverages CoreML, Apple's on-device machine learning framework, to enhance gesture recognition. CoreML offers several advantages: it is optimized for real-time performance, provides seamless integration with widely accessible devices like iPhones, and eliminates the need for bulky external equipment. Compared to conventional gesture recognition techniques, this approach ensures portability, efficiency, and scalability, enabling composers and performers to implement interactive multimedia performances with minimal setup.

The software tools employed in this system are Max/MSP, a visual programming language for music and multimedia developed by Cycling '74, and TouchDesigner, a node-based visual programming environment for creating real-time interactive multimedia content. These tools, combined with CoreML's robust gesture classification capabilities, form the basis of this innovative system.

Ⅱ. Machine Learning Model Design

This study uses machine learning to identify a specific gesture. Machine learning, a field of artificial intelligence, uses computer algorithms that automatically improve through experience. In this study, we chose CoreML, which applies a machine-learning algorithm to a training dataset to create a model, allowing the iOS app to run the models locally.

Fig. 1. CoreML workflow

2-1 Rationale for Selecting CoreML

A machine-learning program using a computer-vision system that receives the motion signal in real-time is required to identify the dancer's motion. It can be implemented using an ordinary webcam or Azure Kinect DK[3],[14],[15]. However, there are some reasons we chose to use CoreML and the iPhone's camera in this study. First, they keep the system simple. If we used an ordinary camera and machine learning, we would need one more program on the computer. In contrast, a camera and machine learning can be integrated into one iPhone. Second, CoreML offers a user-friendly interface, making it accessible even to artists without programming experience. Users only need to organize their dataset into folders, standardize the motion videos by aligning frame rates (30 fps or 60 fps) and durations, and click the machine-learning button to generate a model. This simplicity makes CoreML especially convenient for creative applications.

The last reason is the vast potential offered by mobile phone cameras. Cell phone cameras are advancing rapidly, and a machine-learning model using

a LiDAR sensor or other advanced sensors could be applied in future studies. However, in this study, LiDAR was not used due to performance concerns; it was determined to be impractical for real-time interaction given the processing limitations of current smartphones. The final reason for using this approach is portability. Anyone with an iPhone can operate the system anytime and anywhere, making it highly accessible. Smartphones come equipped with wireless network capabilities, allowing users to interact with the system without spatial limitations. Additionally, smartphones' compact and lightweight design enhances convenience, making this setup ideal for a range of interactive multimedia content where ease of use and mobility are essential.

Table 1. System architecture comparison

Existing System	Camera →	Program: Machine Learning →	Program: Max/MSP TouchDesigner
Current System	iPhone: CoreML →		Program: Max/MSP TouchDesigner

2-2 Rationale for Using Machine Learning

Why use machine learning? Compare previous methods with the data from machine learning. For example, in the motion of raising a hand, what is the difference between mapping data by obtaining coordinates and recognizing a specific gesture using machine learning? Machine learning can distinguish the direction the hand is moving. Of course, one could write a program to distinguish the direction. However, machine learning is simpler and more effective.

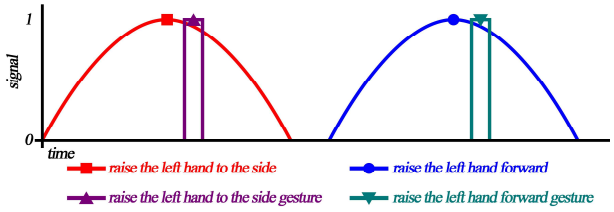


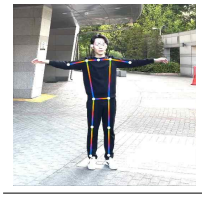
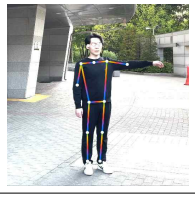
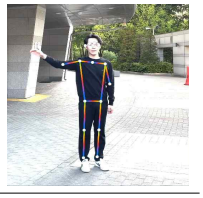
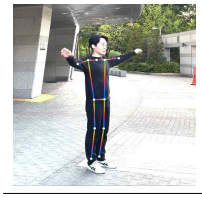
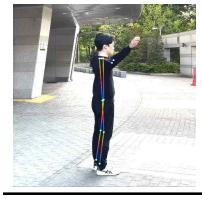
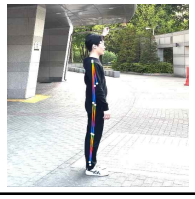
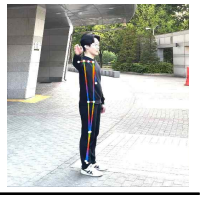
Fig. 2. Comparison between y-axis measurement and machine learning

The change in the y-axis is the same when raising the hand to the side and raising it forward. The result is the same even if the speed value of the movement is obtained. However, machine learning allows the capture of various data by separating the components of the hand or other parts of the body's movement. For example, in conventional one-to-one movement mapping, audio and visuals are changed at half their intended capacity. However, when using machine learning, they are changed based on the correspondence rate, preventing them from being applied at half capacity.

2-3 Dataset for Machine Learning

We had to prepare videos for the machine-learning dataset. This study classified six gestures (raising both hands to the side, raising the left hand to the side, raising the right hand to the side, raising the left hand forward, raising the right hand forward, and no movement). Table 2 shows examples of sideways gestures. Videos taken from various angles were edited at two-second intervals. The gesture of raising the hand forward was organized similarly. There were 40 or more videos per gesture.

Table 2. Examples of sideways gestures from the dataset

Raise both	Raise left	Raise right
		
		
		

2-4 Results of Machine Learning

We built three machine-learning models, each of which went through 300 iterations. The results and graphs are as follows.

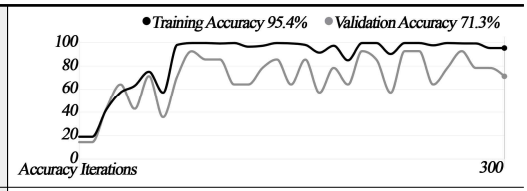
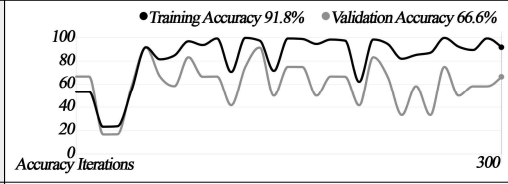
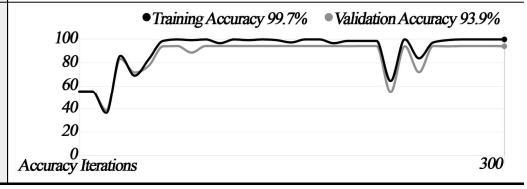
The training accuracy of all the models was greater

than 90%, although the graphs show that Models 2 and 3 found it slightly difficult to distinguish between forward and sideways movement.

Table 3. Gesture classification by models

Model 1	Raise both hands to the side Raise the left hand to the side Raise the right hand to the side Raise the left hand forward Raise the right hand forward No movement
Model 2	Raise the left hand to the side Raise the right hand to the side Raise the left hand forward Raise the right hand forward No movement
Model 3	Raise both hands to the side Raise the left hand to the side Raise the right hand to the side No movement

Table 4. Accuracy graphs by models

Model 1	
Model 2	
Model 3	

Ⅲ. Implementation

We had to build an iOS app to use the machine-learning model. The system in this study was implemented on an iPhone 12 Pro Max. The app has been made publicly available on GitHub [16].

3-1 App Operation

Fig. 3 shows a screenshot of the app in action alongside the Xcode console. To indicate whether motion is being recognized, joint points are highlighted, and the console displays the gesture detected (labeled) and the correspondence rate (confidence level).

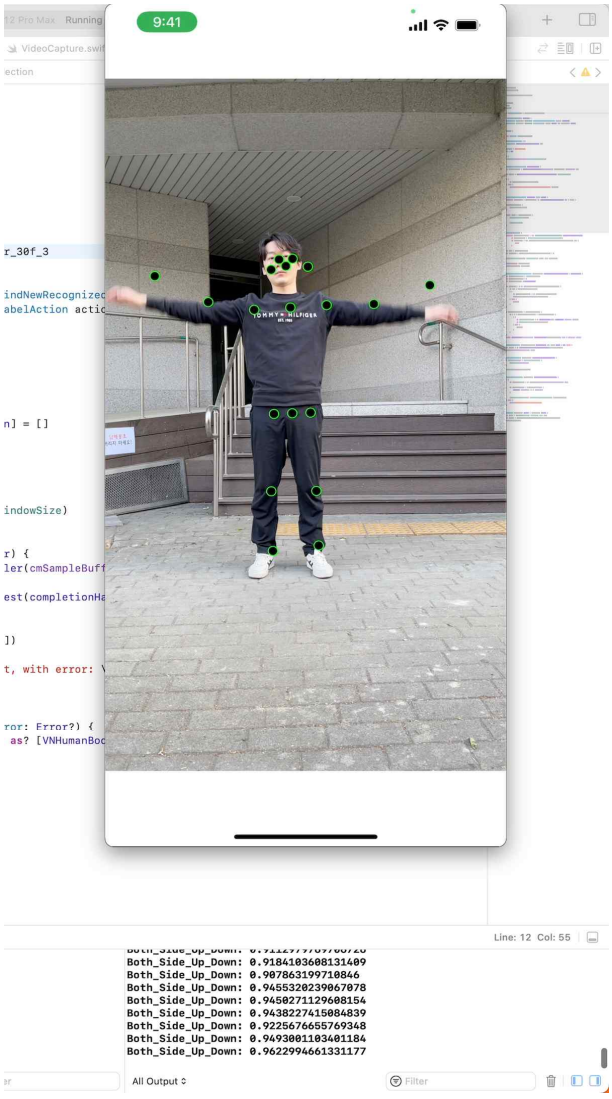


Fig. 3. Screen and console of the app

3-2 Data Capture

The steps in this section were as follows. The gesture was classified on the iPhone. Data identifying the labeled and the confidence level via open sound control(OSC), a protocol for networking sound synthesizers, computers, and other multimedia devices. Therefore, we needed to obtain the current gesture first and then send it to Max/MSP and TouchDesigner via OSC. The confidence level received through OSC was set with a threshold, classifying values above the threshold as 1 and others as 0. Fig. 4 shows a graph obtained by classifying gestures in real-time based on actual movements. The graph on the left displays the confidence level, while the graph on the right shows values converted to on-off signal(1 or 0) based on the

threshold. Red represents “no movement,” yellow represents “raise both hands to the side,” green represents “raise the left hand to the side,” and light blue represents “raise the right hand to the side.” With this system, incomplete gestures or incorrect movements are classified as “no movement.” The system only recognizes and classifies a gesture when it is fully executed.

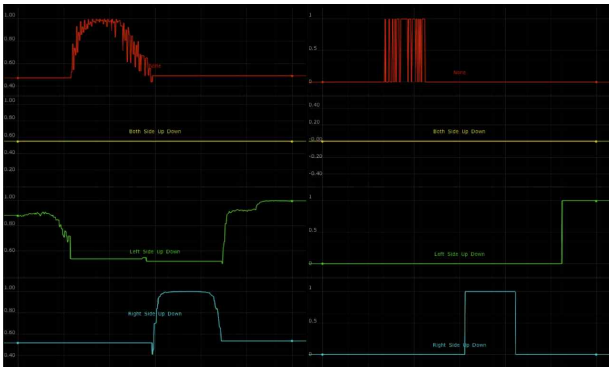


Fig. 4. Real-time gesture classification and confidence level converted to on-off signal graph

Ⅳ. Multimedia Control

In this section, classified gestures are used to control musical instruments and visuals. The instruments were created in Max/MSP, and the visuals were created in TouchDesigner. These patches are also available for download on GitHub[16]. This study presents examples of applying the developed techniques to multimedia content, showcasing how gesture recognition can enable interactive experiences where audio and visual elements respond dynamically to user movements.

4-1 Control of Musical Instruments

We can change the instrument’s timbre by recognizing a specific gesture. In this study, the timbre of the FM synthesizer is altered when both hands are raised sideways. The pitch (carrier frequency) and modulator frequency of the FM synthesizer are fixed at 440 Hz, while the index is modified by the gesture. When we detect the motion of raising both hands sideways, the index of the FM synthesizer increases to 5 over the course of one second and then decreases back to 1. The FM synthesizer patch illustrated in Fig. 5 was developed using Max/MSP. This patch features a section designed to receive OSC signals, which are used to manipulate the index.

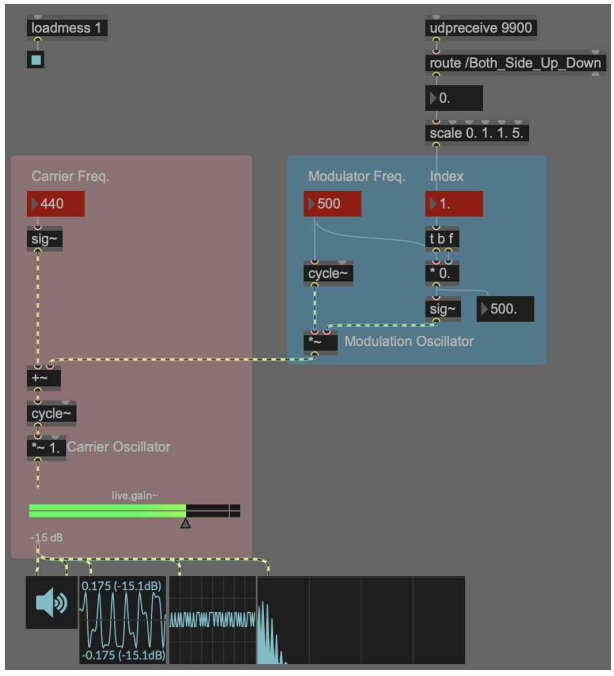
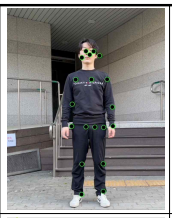
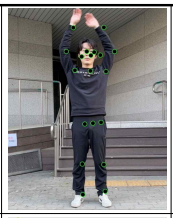
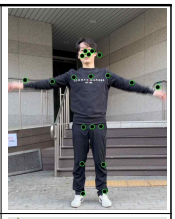
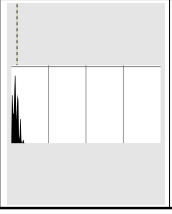
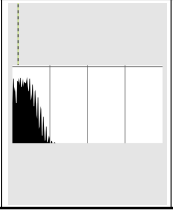
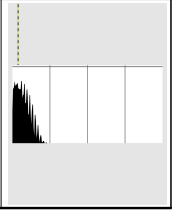


Fig. 5. FM synthesizer patch to control index using gesture

Table 5 illustrates the timbral changes according to variations in movements.

Table 5. Control of musical instruments

Gesture			
Audio			

4-2 Control of Visuals

We can change the visuals when a specific gesture is recognized. In this study, visuals were created using noise, which generates points and connects them with lines based on the distances between the points.

Additionally, a magnet effect, similar to a pulling effect, was incorporated to make the visuals respond more intuitively to gestures. This magnet effect was applied to demonstrate more clearly how the visuals change based on movement, enhancing the overall interactive experience. This system activates only when motion is detected. Similar to controlling the musical instruments, when we detect the motion of raising the left hand sideways, the visuals stretch to the left and then return to their original position. Figure 6 illustrates the patch responsible for generating 3D graphics using a 2D noise image, which also applies a magnet effect based on the movement of the left hand. This system is designed so that elements are connected by lines, depending on their distance from each other, as depicted in the patch shown in Fig. 7.

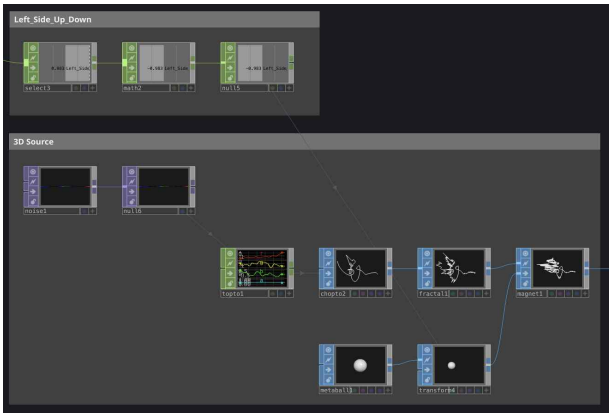


Fig. 6. TouchDesigner patch designed to create 3D sources and implement a magnet effect

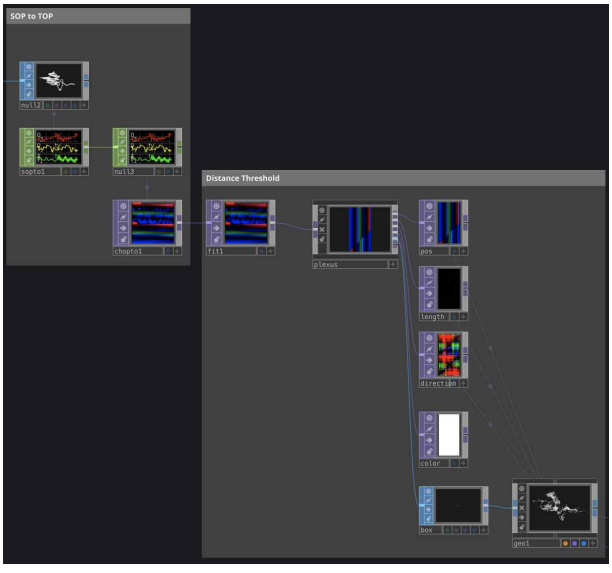
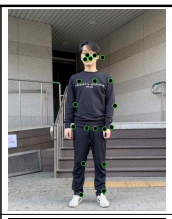
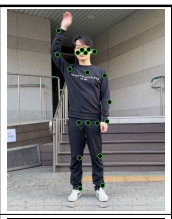
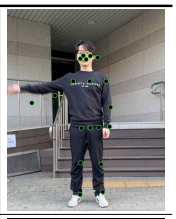
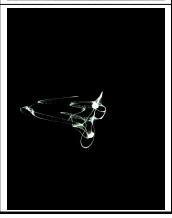
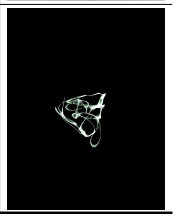


Fig. 7. TouchDesigner patch to connect points with lines according to their distances

Table 6 demonstrates how visual elements shift in response to movements.

Table 6. Control of visuals

Gesture			
Visual			

V. Conclusions

Many contemporary interactive multimedia performances seamlessly blend music, visuals, and dance as artists continually push boundaries to explore innovative ways to perceive and engage with dancers' movements and gestures. While traditional motion capture systems are often used in these performances, we advocate for a shift towards utilizing more accessible technology, specifically the iPhone. By leveraging machine learning for gesture classification instead of relying on conventional one-to-one movement mapping, we can achieve more dynamic, nuanced, and responsive results. This approach fosters a richer interaction between performers and their digital environments, creating an experience that evolves in real-time with each dancer's movement. Looking ahead, this method opens up a thrilling range of possibilities for developing increasingly responsive and immersive multimedia experiences. Gesture recognition technology not only enhances artistic expression but also simplifies the operational aspects of performances. Typically, managing the technical side of live shows and transitioning between segments can be labor-intensive and complex, but by integrating this advanced technology, we can significantly lighten the load on operators.

This study has certain limitations that should be addressed in future research. One key limitation is the reliance on the iPhone's camera, which, while

convenient and accessible, has constraints in terms of field of view, resolution, and depth-sensing capabilities. These limitations can impact the accuracy of gesture recognition, particularly in complex or crowded performance settings. To address this, future work could incorporate LiDAR technology available in some iPhone models, enhancing depth perception and spatial accuracy to make the system more robust and adaptable to diverse performance contexts. Furthermore, while this study focuses on distinguishing simple gestures, future research will explore recognizing more complex and varied gestures and applying these advancements to artistic creations. Efforts will also be made to develop systems capable of recognizing multiple performers simultaneously, enabling the analysis of group dynamics and collaborative movements to influence multimedia outputs. By addressing these challenges and pursuing these directions, we aim to refine this technology, unlocking new creative possibilities and delivering transformative experiences that resonate with both performers and audiences.

Acknowledgment

This work was supported by the Dongguk University Research Fund of 2023.

References

[1] W. Siegel, Dancing the Music: Interactive Dance and Music, in *The Oxford Handbook of Computer Music,* New York, NY: Oxford University Press, ch. 10, pp. 191-213, 2012. https://doi.org/10.1093/oxfordhb/9780199792030.013.0010

[2] R. Behringer, "Gesture Interaction for Electronic Music Performance," in *Proceedings of the 12th International Conference on Human-Computer Interaction (HCI International 2007)*, Beijing, China, pp. 564-572, July 2007. https://doi.org/10.1007/978-3-540-73110-8_61

[3] S. Bhattacharya, B. Czejdo, and N. Perez, "Gesture Classification with Machine Learning Using Kinect Sensor Data," in *Proceedings of the 3rd International Conference on Emerging Applications of Information Technology*, Kolkata, India, pp. 348-351, November-December 2012. http://dx.doi.org/10.1109/EAIT.2012.6407958

[4] International Computer Music Association. A Real-time Platform for Interactive Dance and Music Systems [Internet]. Available: http://hdl.handle.net/2027/spo.bbp2372.2000.138.

[5] L. Deng, H. Leung, N. Gu, and Y. Yang, "Real-Time Mocap Dance Recognition for an Interactive Dancing Game," *Computer Animation & Virtual Worlds*, Vol. 22, No. 2-3, pp. 229-237, April-May 2011. http://dx.doi.org/10.1002/cav.397

[6] J. James, T. Ingalls, G. Qian, L. Olsen, D. Whiteley, S. Wong, and T. Rikakis, "Movement-Based Interactive Dance Performance," in *Proceedings of the 14th ACM International Conference on Multimedia (MM '06),* Santa Barbara: CA, pp. 470-480, October 2006. http://dx.doi.org/10.1145/1180639.1180733

[7] International Computer Music Association. The Digital Baton: A Versatile Performance Instrument [Internet]. Available: http://hdl.handle.net/2027/spo.bbp2372.1997.083.

[8] International Computer Music Association. The "Conductor's Jacket": A Device for Recording Expressive Musical Gestures [Internet]. Available: http://hdl.handle.net/2027/spo.bbp2372.1998.261.

[9] T. M. Nakra, Y. Ivanov, P. Smaragdis, and C. Ault, "The UBS Virtual Maestro: An Interactive Conducting System," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2009)*, Pittsburgh: PA, pp. 250-255, June 2009. https://doi.org/10.5281/zenodo.1177637

[10] T. M. Nakra, "Interactive Conducting Systems Overview and Assessment," *The Journal of the Acoustical Society of America,* Vol. 135, No. 4_Supplement, 2377, April 2014. https://doi.org/10.1121/1.4877850

[11] J. A. Paradiso and F. Sparacino, "Optical Tracking for Music and Dance Performance," in *Proceedings of the 4th Conference on Optical 3D Measurement Techniques*, Zurich, Switzerland, pp. 11-18, September 1997.

[12] Wayne Siegel. Sisters [Internet]. Available: http://waynesiegel.dk/wp-content/uploads/2013/08/Sisters.pdf.

[13] Wayne Siegel. Movement Study [Internet]. Available: http://waynesiegel.dk/wp-content/uploads/2013/08/Movement_Study.pdf.

[14] C.-S. Fahn, S.-E. Lee, and M.-L. Wu, "Real-Time Musical Conducting Gesture Recognition Based on a Dynamic Time Warping Classifier Using a Single-Depth Camera," *Applied Sciences,* Vol. 9, No. 3, 528, February 2019. https://doi.org/10.3390/app9030528

[15] M. Raptis, D. Kirovski, and H. Hoppe, "Real-Time Classification of Dance Gestures from Skeleton Animation," in *Proceedings of the 2011 ACM*

SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '11), Vancouver, Canada, pp. 147-156, August 2011. http://dx.doi.org/10.1145/2019406.2019426

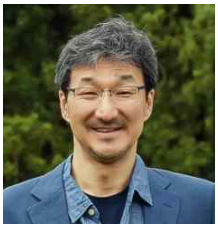
[16] GitHub. Gwangyu-Lee/iOS-Gesture-Classifier-Using-Core ML-with-TouchDesigner [Internet]. Available: https://github.com/gwangyu-lee/iOS-Gesture-Classifier-using-CoreML-with-TouchDesigner.

이관규(Gwangyu Lee)

2022년 : 동국대학교 영상대학원 멀티미디어학과 컴퓨터음악 석사

2024년 ~ 현 재: 동국대학교 영상대학원 멀티미디어학과 강사

※관심분야 : 멀티미디어콘텐츠, 컴퓨터음악, 인터랙티브 퍼포먼스, iOS 애플리케이션, XR



김준(Jun Kim)

1989년 : 경희대학교(음악학사)

1994년 : Boston University(음악석사)

1999년 : Stanford University(음악박사)

2001년 ~ 현 재: 동국대학교 영상대학원 멀티미디어학과 교수

※관심분야 : 멀티미디어콘텐츠, 컴퓨터음악, 인터랙티브 퍼포먼스