A Wearable Action Cueing System for Theatrical Performance Practice

Ryosuke Takatsu[1(✉)], Naoki Katayama[1], Tomoo Inoue[2], Hiroshi Shigeno[3], and Ken-ichi Okada[3]

[1] Graduate School of Science and Technology, Keio University, 3-14-1, Hiyoshi, Minatokita-ku, Yokohama, Kanagawa, Japan
{takatsu,katayama}@mos.ics.keio.ac.jp

[2] Faculty of Library, Information and Media Science, University of Tsukuba, 1-1-1, Tennodai, Tsukuba, Ibaraki, Japan
inoue@slis.tsukuba.ac.jp

[3] Faculty of Science and Technology, Keio University, 3-14-1, Hiyoshi, Minatokita-ku, Yokohama, Kanagawa, Japan
{shigeno,okada}@mos.ics.keio.ac.jp

Abstract. This study focuses on supporting theatrical performances. A system that supports actors learning their acting orders has been already proposed. The system detects actors' speech and cues the next actor to speak. It can however only respond to speech, despite the fact that movement is a very important factor of theatrical performance. Thus this paper proposes to include moving action for cueing the acting order. The evaluation showed that the proof-of-concept prototype system decreased the incidence of both order mistakes and speech mistakes. The findings of this study confirm that the system can efficiently support theatrical performance practice.

Keywords: Acting order · Theatrical performance · Cueing individually · Smart watch · Websocket

1 Introduction

Theatrical performances are a common activity around the world. The theatrical performances are collaborative work, which each person has different role. The most theatrical performances are played according with librettos. Librettos usually have detail instructions, about lines, gesture and who acts it, in timeline format. Actors have to remember a lot of things in addition to the speech lines, including movement, position on-stage, and the flow of the scene. To understand these factors well enough to offer a convincing performance, repeated practice and time are necessary. However, in many case, excessive location costs and schedule coordination make it very difficult to provide enough opportunities to sufficiently learn these elements. Correspondingly, actors compensate for the lack of time and practice with private practices. Actors can refine their own actions like their speech and movement, but it can still be difficult for actors to grasp

T. Yoshino et al. (Eds.): CollabTech 2016, CCIS 647, pp. 130–145, 2016.
DOI: 10.1007/978-981-10-2618-8_11

their relationship with other actors. Consequently, actors have to hold many group practices to understand this dynamic. Making group practice more efficient can hasten the actors' understanding, thereby reducing costs and improving the quality of performances.

Movement plays an important role in theatrical performances. It shows the audience what the actor is doing and feeling in a way that makes theatrical performance the integrated art of speaking and moving. Just as there is an order to the spoken lines, the movements of the performance also have an order. In some scenes, actors sometime move without speaking, which can then prompt other actors to move or speak. Movement is just as important as speaking. To accurately and sufficiently support theatrical performances, it is necessary to focus on the movement as well as the spoken elements. In addition, the movements of a performance have intentions just like the speaking. As a result, mistakes of movement can impact the actors just as mistakes in the libretto.

The biggest problem in practices is the unwanted interruption from fatal mistakes that cannot be immediately repaired. In these situations, mistakes in the acting order represent the most major problems. Even when actors remember own actions well enough, they can easily make mistakes in the acting order if they lose their relationship with other actors. For example, it is not uncommon for actors to mistakenly speak before another scheduled actor, or forget to move appropriately although it is their turn to act. When these mistakes happen, the director stops the practice, confirms the cause, offers a correction, and the actors replay from a previous point in the scene. It is not only a waste of time, but it also causes stress for the actors. To solve these situations, someone takes on the role of "prompter", who is responsible for keeping the acting ongoing. When actors forget their lines or stammer, the prompter immediately displays the lines on boards for the actors to read. This helps keep the acting smooth, and quickly resolves the stress of the situation. Other actors often take on this role in their free time when they are not involved in the scene. In this way, showing critical information at the appropriate time makes practice more efficient.

In the previous research, which focused on helping actors understand acting order [1], specifically targeting group practice situations. The system detects actor's speech, and then cues each actor in the proper order. The cues are different for each actor depending on their roles and the librettos are stored in the system in advance. By practicing with this system, actors can grasp the flow more easily and concentrate on their own actions. That work focused only on speaking performances, but movement is another very important factor in the theatrical performance. By including movement, we can expect the system to comprehensively support the performance. This study evaluated how this system functions and results confirm a decrease in both order mistakes and acting mistakes. These findings confirm that this system can efficiently support theatrical performance practice.

We describe the flow from the first reading of the librettos to the final rehearsal. The purpose of practice is to enhance the quality of performance. In the beginning, actors get together and read the librettos. In this "read-through", the actors each read their own lines, in order. In this step, the actors have the librettos in hands, and they usually sit on a chair rather than practice the movements. This step allows the actors to grasp a rough image of the story. Next, the actors stand up. They actually move around as they speak along the libretto. In this "Run-through practice", the actors confirm their own movements and standing positions. Finally, they have a "rehearsal" right before the recital. In the rehearsal, the costumes, apparatuses, illumination, and sounds are used exactly as they will be used during the recital.

The Run-through practice is composed of three steps: rough run-through step, extract-step, and sometime-stop-step. In the first step, the actors proceed through the performance without stopping. Even where there are mistakes, the actors continue until the last of libretto, except in the case of extreme failure. Basically, the actors do not focus on their librettos, rather the purpose of this step is to roughly understand the entire flow of the performance. In the second step, the director intermittently stops the actors and instructs them to repeat the libretto or their movements. In the third step, the actors proceed through the performance more smoothly, as long as the director does not mind. If the director notices a mistake that must be corrected, the actors are stopped and briefly corrected before they begin again. After the actors proceed through these steps, they usually only engage in Run-through practice, not rough practice. This is based on the premise that the actors understand their own acting well enough by this step and can correct their own mistakes. In this paper, we focus on these Run-through practice steps.

2 Related Works

In the theatrical performance field, many kinds of works have been researched. Kato et al. proposed an automatic scenario-making system using the information about character and things that have been drawn in preliminary pictures [2]. Sugimoto et al. proposed a scenario making system called "GENTORO" which uses robots and a handy projector [3]. Additionally, there are animation systems called "Pixel Material" [4] for children. In this regard, there is much research supporting story creation.

Much work has also been done on staging. For example, there are systems for presentation apparatuses and illumination on the PC [5], as well as presentation renditions for actors [6]. Kakehi et al. proposed "Tablespace Plus" with which a user can interact with stage information by moving objects [7] A derivative application focuses on the actors' standing positions. In addition, there is a system that deals with 3-dimensional position information [8]. In most of the systems that support staging, computer graphics are used.

In terms of performance planning support systems, there are many types of software being sold. "WYSIWYG" by CAST company [9] focuses on setting stage illumination and spotlights. "Matrix" by Meyer Sound company [10] supports sound needs. Furthermore, there is some research for multi planning collaboration in face-to-face [11] and remote situations [12]. Geigel et al. proposed natural and intuitive interfaces for virtual theatrical performances [13].

Although not specifically designed for theatrical performances, there is the presentation system for a chairperson, activated by using a wearable device. However, this system is only for one chairperson, and not for multiple users.

As described above, many people work to support theatrical performances. However, we cannot find one that focuses on the ordering of moving action in the theatrical field. Moreover, in the field of collaborative work, the same is true.

3 Cueing Speeches and Action Individually

3.1 The Cueing Acting Order System for Theatrical Performance

This study proposed a system of hands-free, wearable devices that individually cues speaking and movement order for each actor. The vibrations of the wearable devices signal the cues, so the actors do not have to see the device's display to know the order. The system sends cues based on the libretto, and provides instructions for each actor individually. In addition, the condition for progress are the actors' speech and movements, so the system clock can measure the relative time, not the absolute time. Both speech and movement factors are incorporated together in order, so this system can comprehensively support the performance. Furthermore, the system can tells when actor makes mistakes in movement, so actors can confirm own their movements. By using this system, actors can concentrate on their own acting and learn the flow in the early stages of theatrical practice.

3.2 Actors' Action Management

Actors can learn and improve their own lines and movements for each performance. However, acting in a relationship with other actors is difficult to understand through solo training. We decided to take libretto in the system to better manage the actors' order of speaking parts. This allows the actors to learn their parts without fully understanding the actions of the other actors. Actors just act when they are cued by system, which enables them to act in the appropriate order every time.

3.3 Efficiency of Run-Through Practices

Mistakes in the acting order happen frequently and can have a huge impact on practice. It stops Run-through practice and applies more stress to the actors, so it is important to make Run-through practices more efficient. Providing better support for Run-through practices will actors to experience the flow of acting more easily.

3.4 Hands Free and Individual Cueing

In the Run-through practice, actors confirm their movements and standing positions. Having to hold the libretto in their hands as they do this creates a hindrance for appropriate training, especially movement practice. Furthermore, each actor has a different role, so they speak different lines and move around the stage differently. As a result, each actor needs different, unique information. To improve Run-through practice, it is necessary to present the appropriate information for each actor individually in a hands-free way. As a good theatrical performance requires the timing of actors' to be synchronized with the movement and libretto of other actors, any system for improving Run-through practice must have a master clock to which all actors' cues are set.

4 Implementation

As a prototype, the proposed system was implemented for three actors. This chapter describe this implementation.

4.1 Hardware Configuration

The system uses WebSocket (described in Sect. 4.2) to connect to a PC server, and Samsung Gear Live as wearable, watch-like devices that can recognize voice and vibrate (Fig. 1). In this system, each actor wears this smart watch to receive acting orders through a vibration and detect the actor's speech with voice recognition.

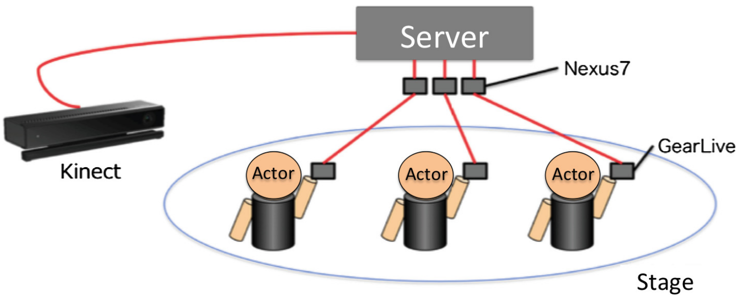

Fig. 1. System configuration

The Samsung Gear Live smart watches are controlled using Nexus 7 tablets. One tablet can control one smart watch, so this prototype system used three tablets connected through Bluetooth. Each table has to be near the actors, but they do not have to wear or carry them. The smart watches use Android Wear 5.0.1 as the implement platform and Android Java API v21.1.2. The processor is a 1.2 GHz Qualcomm Snapdragon 400 and the display is a Super AMOLED

1.63-type. The OS of the tablet is Android Wear 4.4.3, with a 1.3 GHz NVDIA Tegra 3 mobile processor, and an LED backlight 1,280 * 800 type-7 (WXGA) display.

Kinect for Windows API v2.0 was used as a depth camera to recognize each actor's movements. It has a depth camera, an RGB camera, a multi-eye ray-microphone, and a processor. It can recognize gestures to detect the user's position and height. The system used the "Visual Gesture Builder" as a tool for identifying each actor's movements with machine learning. First, we moved in front of the Kinect and taught the system the movements with Kinect Studio. Second, we labeled the data and set threshold parameters. Finally, we confirmed the accuracy of discriminator. After that, the system outputs true or false for each movement that the user acts in front of the Kinect. We can enhance the accuracy of machine learning to use this process for many people, especially those of various heights. This prototype used three people as a model for this process.

4.2 Connection Environment

To connect quickly and support multi devices, the prototype system used WebSocket, which can keep the socket open in a bidirectional connection. In addition, it supports stable connections in a multi-device environment. This was implemented with node.js and shared with the devices using GitHub and heroku.

4.3 System Function

Connecting Devices. To begin, the server was started and set up to allow access to http://dry-everglades-7373.herokuapp.com. Next, the Kinect (with control PC) and tablets were connected to server. To start program in Kinect and tablets' application, these can be connected to server automatically. IDs are distributed for each device randomly, and each device is manually registered on the server. Next, the tablets are paired to the smart watches using an existing application (Fig. 2). In summary, the procedure for preparing the system is as follows:

1. Accessing the URL (starting server).
2. Starting Kinect program and tablets' app (connecting to server)
3. Pairing tablet and smart watch.
4. Starting smart watch's app.

Management of Actors' Action. The server-control screen is shown in Fig. 3. The librettos were stored as CSV files on the server, which can process and display the data. The relevant libretto information includes the order (number) as well as the corresponding actor, line, and movement. The system can process and transmit three kinds of patterns: line only, movement only, and both line and movement. In cases where actors act at the same time, the system has the same

Fig. 2. Tablet screen for controlling smart watch

order number for these actions and subsequently processes them simultaneously. The corresponding libretto can be matched to the appropriate devices such that the details are displayed on the server and the tablets display each actor's action detail. Until the practice begins, the smart watches display each actor's initial state.

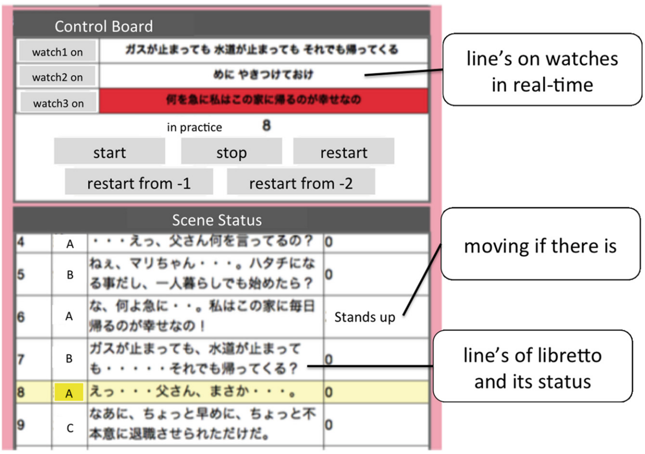

Fig. 3. Server-control screen

Cueing Order for Actors. This system cues acting order for actors using the vibration function of the smart watches. Actors receive information about speaking and movement, so it is necessary for actors to distinguish speaking cues from movement cues. To achieve this, the system uses two vibrations. The smart watch vibrates one time when the actor does not have to move, or twice when actor should move. When cueing movement, the first vibration lasts 500 ms and the second is 700 ms.

The display of smart watch also changes when cued. Although basically stand-by screen is displayed, as shown in Fig. 4(1), when it is cued, display changes to lines screen as shown in Fig. 4(2). If the cue is a movement cue, lines are obviously not shown. This function acts like a prompter and the lines screen changes to the voice input screen within two seconds. The timing is delicate and difficult to judge systematically, so the system allows the actor to freely determine the acting timing.

Fig. 4. Screens of smart watch. 1. Stand-by screen; 2. Lines screen; 3. Voice input screen; and 4. Input result screen

Detection of Speaking and Moving. To cue properly, the system must know the scene the actors are performing in real-time, which it does by detecting both speech and movement. Speech is detected using the smart watch's voice input function, and movement is detected by the depth camera of the Kinect.

After cueing an actor, the smart watches start detecting voice, which is implemented in RecognizeIntent in the Android.Speech package. However, the voice recognition process starts after finishing speech, it takes a moment for the voice input to be recognized.

When the smart watch sends a cue, it displays the lines screen for a short time accompanied by a vibration; two seconds later, it shows the voice input screen. Even when the lines screen is displayed, voice detection has already started. The smart watch then sends the results of this voice detection to the server. The smart watches only perform voice detection for the actor who should be speaking, so the watches worn by other actors are not activated.

To detect voice, the smart watches do not have to be near the actor's mouth. These smart watches can detect voice even if there is certain distance from the mouth, like the natural location of the wrist at the side of the actor's body.

To detect movement, the system used the skeleton detect function of Kinect. Figure 5 shows the PC screen when detecting an actor's movement. This works only in scenes when actors have to move. Kinect can detect max 6 users and it's coverage is max 70° and 4.5 m depth.

Fig. 5. Kinect detecting users

Collation of Detection Result and Notification. If the result of detection is right, the system proceeds to next scene. When recognizing speech, the system only judges whether the actor who is speaking is right, but does not judge the words for accuracy. As long as actors follow this system and do not act without cueing, the actors will not miss the acting order.

The system used the Visual Gesture Builder to recognize gestures. Each actor's movement data is registered in the system in advance. If the actor's movement nearly matches the registered movement, Kinect send a True message to the server.

When actors fail the movement or do not move with seven seconds of the time they should move, the system sends a sound notification.

4.4 Actors' Flow

1. Receive cue by vibration
2. Acting
3. If fail, receive notification by sounds

The actors' process is very simple. The actor receives a cue through a vibration, then performs the speech or movement. If the actor acts properly, the next actor receives the next appropriate cue; if the actor fails, the system stops the practice and the smart watch sounds a notification. To restart the practice, the actor touches the buttons on smart watch display. For this purpose, the smart watch has three buttons: stop, back one step, and back to the beginning, of which the last two can only be used in stop state.

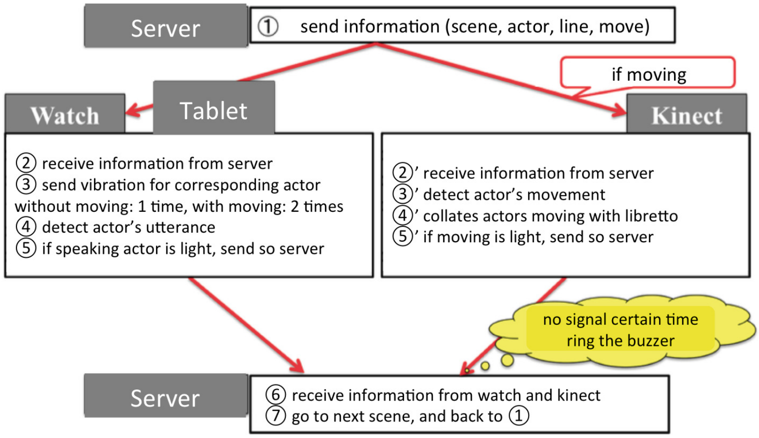

Fig. 6. System flow

4.5 System Flow

The system flow is shown in Fig. 6. For cueing an action to an actor, the server sends the appropriate data, including the order, the actor who should act, the lines, and the movement to each tablet device and Kinect. If the acting has no movement, the server does not send data to Kinect. Next, each smart watch receives the data from the tablet. If the "actor who should act" coincides with the actor registered in the watch, it vibrates. If there is no movement, it vibrates one time; otherwise, it vibrates twice. Then the smart watch changes its mode to detect voice. When the actor speaks, the system determines whether the speaker is light, in which case the smart watch sends "true" to server. If the actor should move, Kinect receives data from the server, and starts recognizing movement. If Kinect finds that "actor who should act" acts light "movement", Kinect sends "true" to server.

The server receives the results from the smart watch and Kinect. If the server sent data to Kinect and does not receive "true" back from Kinect, the server determines that the actor failed to move, and sounds the buzzer. If the server receives true, it proceeds to send data about the next action to each device. This flow continues until the end of the libretto.

An example of a practice timeline is shown in Fig. 7, where the horizontal axis represents time. The colors shown on the right correspond to the line colors in the figure, and white lines represent when the system detects voice or movement. In this example, the flow is as follows:

1. Actor A: speech and movement
   (a) Smart watch vibrates twice.
   (b) Smart watch shows lines and starts voice detection. Kinect starts recognizing movement.
   (c) Smart watch detects speech and Kinect recognizes movement.
   (d) Kinect sends "true" to server.
   (e) Smart watch finishes detecting and sends "true" to server.
   (f) Server goes to the next action.
2. Actor B: only speech
   (a) Smart watch vibrates once.
   (b) Smart watch shows lines and starts voice detection.
   (c) Smart watch detects voice and sends "true" to server.
   (d) Server goes to the next action.
3. Actor C: only movement
   (a) Smart watch vibrates twice.
   (b) Kinect starts recognizing movement.
   (c) Kinect recognizes movement and sends "true" to server.
   (d) Server goes to the next action.
4. Actor A and B: only speech
   (a) Each smart watch simultaneously vibrates once.
   (b) Smart watch shows lines and starts voice detection.
   (c) Smart watch detects voice and sends "true" to server.
   (d) Server goes to the next action.
5. Actor C: speech and movement (mistake)
   (a) Smart watch vibrates twice.
   (b) Smart watch shows lines and starts voice detection. And Kinect starts recognizing movement.
   (c) Smart watch detects speech, however Kinect does not recognize movement.
   (d) Smart watch finishes detecting and sends "true" to server.
   (e) Server does not receive "true" from Kinect, so server determines it has encountered a mistake and sounds buzzer.
   (f) Pause the practice.

5 Evaluation

This chapter provides an evaluation describing whether this system can effectively support theatrical performance practice.

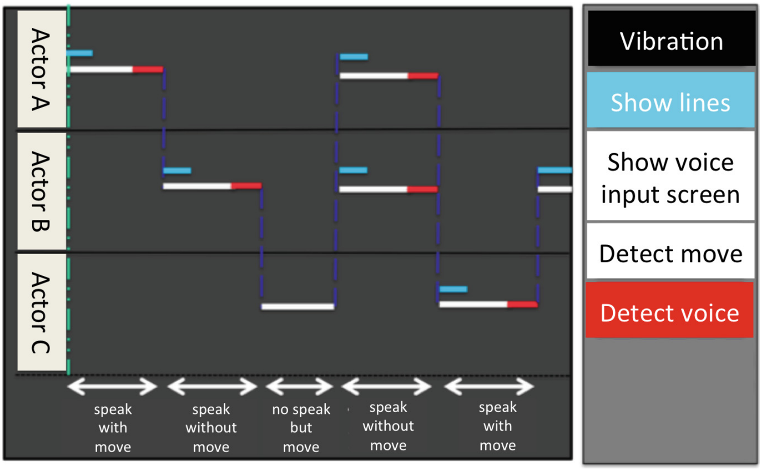

Fig. 7. Timeline of practice with the system (Color figure online)

5.1 Outline

Purpose. The purpose of this analysis is to evaluate whether the system decreases mistakes and facilitates practice.

Method. Participants used the proposed system in actual practice, acted according to the librettos prepared for them, and the number of mistakes of speech and movement were counted.

Participants. This experiment included 24 university and graduate school students, all of whom were novices in theatrical performance. The participants were divided into eight groups of three.

Comparison. The system measures presence or absence. Kinect's recognition and sound buzzer were used to evaluate the effects of cueing, in both environments. The number of mistakes were counted in both environments.

Evaluation Items. This experiment evaluated four types of mistakes: speech order mistakes, speech line mistakes, movement order mistakes, and movement content mistakes. Regarding speech line mistakes, the experimenter allowed actors to make small mistakes as long as they did not change the meaning, and the actors were not required to speak the exact words in the libretto. The number of movement mistakes was compiled with the Kinect's buzzer count.

Librettos. This experiment used librettos, both with three actors, each with 12 – 14 speaking parts and 5 – 7 movements. These librettos were called "Libretto A" and "Libretto B".

Questionnaire. After the experiment, the participants completed a simple questionnaire with the follow questions:

– Which system did you feel easy to practice?
– Please tell us the reason.
– Please score own acting. (from 1–5 for speech and 1–5 for movement)
– Please include other relevant comments.

Procedure. Based on the advice of actual theatrical performers, this experiment set the time for remembering the libretto.

1. Explaining the experiment and how to use the system to the participants.
   First, we explained the experiment. Next, to help the actors become familiar with the system, we explained how to use the system and its flow. Then, we explained that the smart watches cannot detect soft voices and that participants should speak clearly. We also evaluated how the participants remembered the libretto in this experiment, so we told them not to watch their smart watch displays.
2. Participants memorize the libretto in private.
   Roles were determined and the libretto was distributed to the actors, who were then given five minutes to memorize their lines.
3. Read-Run-through practice.
4. Experimenter teaches the participants their movements.
   The libretto also included instruction for movement. However, in this system, Kinect cannot recognize slight differences in the interpretation of the actual movements, so the actors were shown how to move accurately.
5. Participants memorize libretto in private again.
   The actors were given three minutes to memorize the movements that should accompany their lines.
6. The experiment in Run-through practice.
   Participants performed the Run-through practice experiment and the number of mistakes were counted. If participants failed, the practice was paused while they were instructed, and then restarted from the failure point. Participants tried this twice.
7. Repeat steps 2–6 with a second libretto.

Pattern of Experiment for Each Group. To offset the effect of order difference, the experiment proceeded according to the process shown in Table 1.

5.2 Result

The results of this experiment are shown in Fig. 8. The bars in this graph represent the average per Run-through practice. The t test results are as follows:

Table 1. Example of pattern for experiment

	Group 1	Group 2	Group 3	Group 4
Take 1	With system Libretto A practice twice	Without system Libretto A practice twice	With system Libretto B practice twice	Without system Libretto B practice twice
Take 2	Without system Libretto B practice twice	With system Libretto B practice twice	Without system Libretto A practice twice	With system Libretto A practice twice

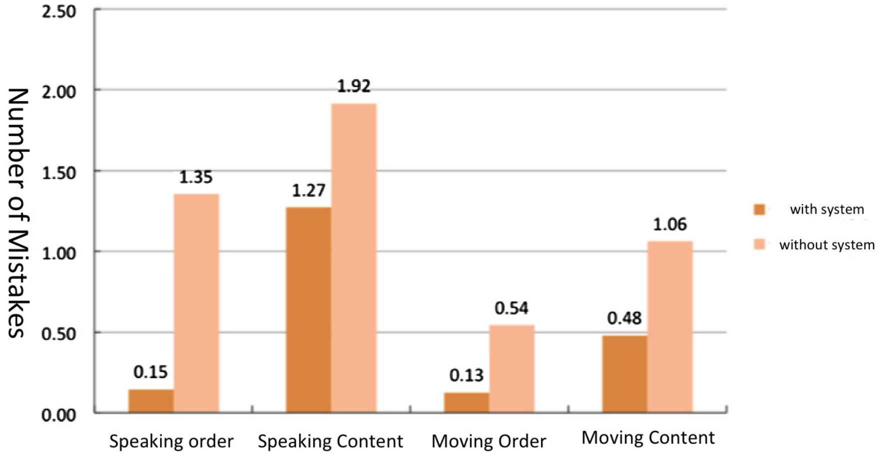

Fig. 8. The result of number of mistakes

- Speaking order mistake: $t(47) = 5.58$, $p < 0.05$
- Speaking lines mistake: $t(47) = 3.05$, $p < 0.05$
- Moving order mistake: $t(47) = 4.19$, $p < 0.05$
- Moving content mistake: $t(47) = 3.52$, $p < 0.05$

We confirmed significant difference in all items, and the standard deviation is shown in Table 2. We also confirmed whether we could offset the effects of the experimental order:

Table 2. The results of standard deviation

	Speech order	Speech intent	Moving order	Moving
With system	0.456	1.604	0.331	0.763
Without system	1.534	1.835	0.815	0.852

- Difference of librettos: $t(7) = 0.24$, $p > 0.05$
- Difference of experiment order: $t(7) = 0.46$, $p > 0.05$

These results confirm that we could offset the effects of the experimental order.

5.3 Discussion

The results of the experiment demonstrate that the system can decrease mistakes, so using this system in Run-through practice would allow actors to practice more efficiently. Specifically, this experiment showed a decrease not only in order mistakes but also in line and movement mistakes. Although our system just cues the order (participants did not watch the lines screen), it has an effect on lines and movement content. This means that participants could concentrate on remembering content and not be distracted by remembering acting order. This was noted in some of the comments in the post-experiment questionnaire, which also confirmed a large difference in self-assessment. Participants can act with confidence using the cueing order prepared by the system. These results clearly show that this system can accurately and efficiently coordinate theatrical performance practice in real time.

6 Conclusion

Actors in a theatrical performance are forced to memorize a lot of information in the libretto. To grasp the acting order, actors must understand not only their own acting but also their relationships with other actors. The proposed system helps actors learn to act in the proper order by cueing the order for each actor individually. However, movement is also a very important factor in theatrical performances, so this study extended the functionally of the system to focus on the actor's movements in addition to their speaking parts. This experiment confirmed that using this system lead to a decrease in the number of order mistakes and acting mistakes during Run-through practice. This system allows actors to more fully concentrate on their own acting and grasp the acting flow in the early stages of the practice. Overall, this system was effective in helping actors improve their performance during Run-through practice.

References

1. Ryosuke, T., Naoki, K., Tomoo, I., Hiroshi, S., Ken-ichi, O.: A wearable real-time cueing system for theatrical performance practice. In: 13th IPSJ SIG Digital Contents Creation, No. 8 (2016)
2. Shigeru, K., Takehisa, O.: The support system for story creation using pictures. In: Proceedings of the 2006 International Conference on Game Research and Development, pp. 141–148. Murdoch University (2006)
3. Masanori, S., Toshitaka, I., Tuan, N.N., Shigenori, I.: Gentoro: a system for supporting children's storytelling using handheld projectors and a robot. In: Proceedings of the 8th International Conference on Interaction Design and Children, pp. 214–217. ACM. New York (2009)

4. Tal, D., Michal, R.: Pixel materiali: a system for creating and understanding pixel animations. In: Proceedings of the 6th International Conference on Interaction Design and Children, pp. 157–160. ACM, New York (2007)
5. Matthew, L.: Bowen virtual theater. In: ACM SIGGRAPH 2003 Web Graphics, p. 1. ACM (2003)
6. Mel, S., Howell, J., Steed, A., David-Paul, P., Maia, G.: Acting in virtual reality. In: Proceedings of the Third International Conference on Collaborative Virtual Environments, pp. 103–110. ACM, New York (2000)
7. Yasuaki, K., Takeshi, N., Mitsunori, M.: Tablescape plus: Interactive small-sized vertical displays on a horizontal tabletop display. In: Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems, TABLETOP 2007, pp. 155–162. IEEE (2007)
8. Hanyuool, K., Issei, T., Hiroki, Y., Takayuki, K., Satoshi, M., Takeshi, N.: Mario: mid-air augmented realityinteraction with objects. In: Reidsma, D., Katayose, H., Nijholt, A. (eds.) ACE 2013. LNCS, vol. 8253, pp. 560–563. Springer, Heidelberg (2013)
9. Cast Group of Companies Inc. http://www.cast-soft.com/wysiwyg/overview
10. Meyer Sound Laboratories Inc. http://www.meyersound.com/
11. Daiki, N., Yousuke, H., Tomoo, I., Kenichi, O.: Diamond theater: a system for supproting creative activities. Inf. Process. Soc. Jpn. Jornal **51**(12), 2396–2408 (2010)
12. Christian, D., Denis, L.: Avatar: a virtual reality based tool for collaborative production of theater shows. In: The 3rd Canadian Conference on Computer and Robot Vision, p. 35. IEEE (2006)
13. Joe, G., Marla, S., David, H., Namco, B., Brian, J.: Adapting a virtual world for theatrical performance. Computer **44**(12), 33–38 (2011)