Gesture Recognition for Adding Sound Effects

Dan Wang
Computer Science, University of Waterloo
Waterloo, Canada
d97wang@uwaterloo.ca

Wanqi Li
Computer Science, University of Waterloo
Waterloo, Canada
wanqi.li@uwaterloo.ca

ABSTRACT
The task of adding sound effects for stage play is a repetitive task once the design of sound effect is decided. In this article, a gesture recognition system is proposed to automatically reproduce the sound effect as designed by sound designer during rehearsal of the play. The proposed system aims to achieve three properties: transparency, customizability, and reproducibility, in order to minimize the effort needed for the user to adopt the proposed system and to achieve accurate sound effect timing that is robust enough for artistic performance. The proposed system is tested with multiple actors on stage performing kongfu actions. The results suggest that the proposed system has high accuracy in gesture recognition and timing control, even when the number of training samples is limited.

Keywords
Gesture recognition; sound effect; skeleton extraction; Dynamic Time Warping; stage play

1. INTRODUCTION
In a stage play, sound effects are often operated by human sound designer. The sound effects are pre-decided by the sound designer during pre-production. Also, the sound designer has to decide the precise moment to trigger the sound effect. Because of the dynamic environment of stage play and the complexity of adding sound effects, extreme focus is required for the sound designer during the play. Furthermore, the actors and the sound designer usually need to repeat the plot many times during rehearsal in order to achieve perfection of performance. The task of a sound designer tends to be repetitive and laborious, once the design of sound effects is completed. As usual, such repetitive task implies a chance for the machine to replace human labor for lower cost and higher robustness.

Consider the following scenario. In theater, a drama is performed. The mom slapped her daughter. Though there is no physical contact between the actors, a sound effect is played just on time. The sound effect is triggered by the camera, connected to a gesture recognition classifier, which recognizes this action as “slap”. And this classifier has been trained during the rehearsal by the sound designer, so it knows the actors and this act perfectly.

Such a sound effect system is proposed in this article. There are three properties we aim to achieve in the proposed sound effect system: transparency, customizability, and reproducibility, each of which will be explained in due time.

2. RELATED WORK
The proposed system consists of three main components: skeleton extractor, gesture recognizer and key point detector. In this section, we review some related techniques that are applied in the proposed system.

2.1 Skeleton Extraction
Skeleton extractor performs dimension reduction on video stream, to obtain the skeleton of the actor represented as just few 2D points (Figure 1). This greatly simplifies the later recognition problem, as raw video could easily contains millions of pixels, while each skeleton frame consists of just tens of real values.

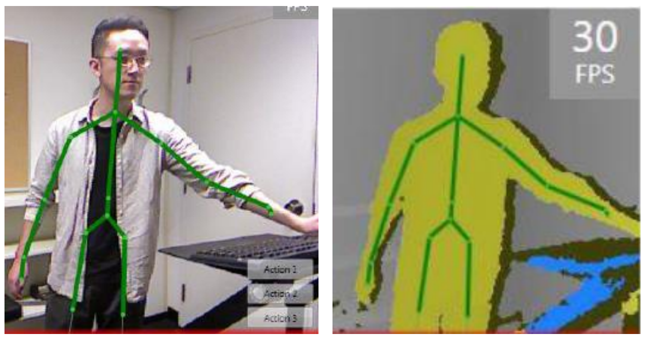

Figure 1. Skeletal joints extracted from depth image

A real-time skeleton extraction scheme [1] is published by a team of Microsoft Research, led by Shotton et. al. Random forest is trained on depth images to estimate the probability of a pixel belonging to certain body part. This way, body part is represented as a probability distribution over pixels. The proposed algorithm is parameter-tuning free and uses no temporal or kinematic information, that is each frame and joint is treated independently. The algorithm is real-time and can extract skeleton data under 5ms per frame. The algorithm works for various poses, body shapes and clothing, and is implemented in Microsoft’s commercially available sensor product Kinect [4].

Depth-Image

The color and texture of a person greatly vary by clothing, hair and skin, not to mention lighting. Therefore color and texture provides little information to distinguish body parts. Such redundant information should not be used to train classifier.

Depth image on the other hand, only captures the necessary geometric information. Each pixel of a depth image is the measured distance of that point to the camera. It describes the front of a 3D shape. We believe depth image contains all the information it needs to distinguish body parts.

Classify a Single Pixel

In [1], geometric information surrounding a pixel on depth image is captured by difference features in the form

$$f_\theta(I, \mathbf{x}) = d_I\left(\mathbf{x} + \frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{v}}{d_I(\mathbf{x})}\right)$$

where I is the depth image, $\mathbf{x}$ is the location of a pixel, d_I denotes the depth at certain location, $\mathbf{u}$ and $\mathbf{v}$ are any specified offset from the given location $\mathbf{x}$.

Many features of the above form is generated to train a random forest [13] on images with tagged body parts. As a result, each body part is represented as a distribution over pixels, which is obtained by averaging the likelihood of a pixel belonging to certain body part according to each decision tree in the forest.

Joint Position

The joint position of each body part is proposed as the mode of a density function define below.

$$f_c(\hat{\mathbf{x}}) \propto \sum_{i=1}^{N} w_i \exp\left(-\left\| \frac{\hat{\mathbf{x}} - \mathbf{x}_i}{b_c} \right\|^2\right)$$
$$w_i = P(c \mid I, \mathbf{x}_i) \cdot d_I(\mathbf{x}_i)^2$$

where c indexes each body part, b_c is a learnable bandwidth parameter indicating the spread of the body part.

These joint positions can be used to represent the pose of the person with just tens of real values.

2.2 Gesture Recognition

The gesture recognizer needed in the proposed system should be real-time and robust with limited number of training samples, since the sample that are automatically annotated by the input of the sound designer during rehearsal and not many rehearsals are expected. There are many off-the-shelf classification algorithms applicable. This report considers the popular Kernel Nearest-Neighbor [2] with Dynamic Time Warping kernel [3].

2.2.1 Kernel Nearest-Neighbor

The "kernel trick" that originates in support vector machine (SVM) has turned out to be useful far beyond SVM and parametric methods. As an instance, in [2], kernel trick is applied to the nearest-neighbor, arguably one of the simplest classification methods. Nearest-Neighbor by default is a non-parametric method, which assumes no prior knowledge in the classification problem. By replacing the distance metric with any kernel [2], the user can impose prior assumptions about the problem in the kernel to achieve better accuracy.

The number of nearest neighbors k affects the accuracy of classification. In this report, k is selected following a procedure very similar to that in [6]. More sophisticated methods such as hyper-parameter optimization [7] involving various heuristics are also considered.

2.2.2 Dynamic Time Warping

Dynamic Time Warping (DTW) [3] is a popular method to compute similarity between two time series. DTW has the property that it is insensitive to time deformations. That is the speed and acceleration of the event does not affect the result. This is achieved by finding an optimal "warped" mapping between the two time series. DTW has been widely used in pattern recognition of time-dependent signals [5].

DTW compares two time-dependent sequences $X = (x_1, x_2, \cdots, x_N)$ of length N and $Y = (y_1, y_2, \cdots, y_M)$ of length M. $C(n, m) = c(x_n, y_m)$ is the cost matrix for the two sequences, where $c(x_n, y_m)$ is the difference between x_n and y_m. Then the goal is to find an alignment between X and Y with minimum cost.

Algorithm of dynamic programming is used to find an optimal warping path that satisfies three constraints [3]: boundary condition, monotonicity condition, and step size condition. The optimal path $p^* = (p_1, p_2, \cdots, p_L)$ is computed in reverse order of the indices starting with $p_L = (N, M)$, until $(n, m) = (1, 1)$ is reached. Suppose p_l is computed, p_{l-1} is updated as follows

$$p_{l-1} := \begin{cases} (1, m-1), & \text{if } n = 1 \\ (n-1, 1), & \text{if } m = 1 \\ \arg\min C(n, m), & \text{if otherwise} \end{cases}$$

Note that the p_{l-1} must be one block closer to $(1, 1)$ than p_l. Compared with the Euclidean way of comparing two series, DTW aligns the two time series first before comparing.

In order to better suit different use cases, variations of DTW are proposed. For example, the basic DTW is vunarable to "singularities", in which case it generates inaccurate mapping. Eamonn J. Keogh et al. [8] proposed a modification on DTW that fixes such problem.

DTW is used by Gavrila et al. in [5] for human movement tracking. Preprocessing [9] is performed on time series to remove certain noises that can affect the performance of DTW. Suranjith De silva et al. [6] carries out a series of experiments with different variations of DTW under different parameter configurations. The result suggests that parameter configuration matters on the performance of gesture recognition and certain DTW variations have superior performance for certain gestures.

2.3 Interactive Machine Learning

The prerequisite of adopting machine learning algorithms to pattern recognition is good examples. In order to improve the quality of input example, Jerry A. et al [12] proposed an interactive machine learning model. This model can interactively generate training samples while carrying out feature selection on the run, so feedback can be returned to the user immediately.

Based on gesture recognition, there are some works on automatically music controlling and producing for gestures or dances, for example, the EyesWeb project [10]. Some researchers also focus on interactive dance performance feedback by producing real time visual and audial feedbacks [11]. These works are related to but different with the problem of adding sound effect to actions in a stage play. For adding sound effect in a stage play, the learning scheme need to pin-point the moment the sound effect to be triggered. Because asynchronous sound effects can easily ruin the aesthetics of the show.

3. SYSTEM DESCRIPTION

There are three properties we aim to achieve in the proposed sound effect system: transparency, customizability, and reproducibility.

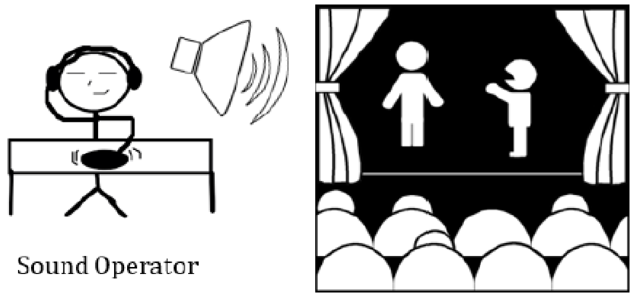

Figure 2. During rehearsal sound effects are added by sound designer

During rehearsal, sound designer operates on the standard sound effect interface, where all his operations are automatically recorded as annotations to the happening actions on the stage (Figure 2). The annotated actions are then used to train a gesture recognizer. Such training process requires minimum attention from the actors and the sound designer, all they should do is just the same as what they usually do in previous rehearsal processes, in this sense it is **transparent**.

Once the system is properly trained, the sound designer can switch the system to recognition mode, during which sound effects will be triggered by the gesture recognizer (Figure 3). Because the training samples are collected from specific actors in the rehearsal, the trained system recognizes and only recognizes actions performed by the same actor in the rehearsal. The trained system is fully **customized** for the actors during the rehearsal.

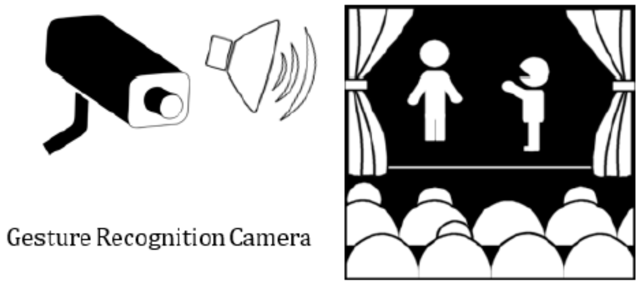

Figure 3. After the system is properly trained, sound effects are automatically triggered.

When the training samples are collected, the timing of the sound effect is also recorded and pin-pointed in the sample action. The system is trained not only to recognize the action, but also to **reproduce** the timing of the sound effect.

The proposed system can be switched back and forth between two modes: rehearsal mode and recognition mode. In rehearsal mode, video stream is converted into skeleton stream. The recorded skeleton data is automatically annotated by the operations of sound designer. The annotated samples are used to train a gesture recognizer and key point detector. Note that, the rehearsal mode requires no extra input from the cast, as the system automatically collects annotated samples for training.

Once the system is properly trained, recognition mode can be turned on. The system applies the trained gesture recognizer to the skeleton stream. If certain registered action is recognized, the trained key point detector will determine a time to trigger the sound effect. The recognition mode adds sound effect to actor's actions automatically. Sound designer does not need to do anything once recognition mode is on.

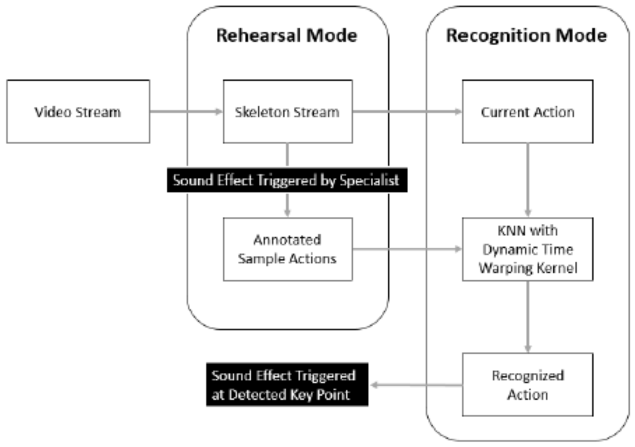
Rehearsal Mode
Recognition Mode
Video Stream
Skeleton Stream
Current Action
Sound Effect Triggered by Specialist
Annotated Sample Actions
KNN with Dynamic Time Warping Kernel
Sound Effect Triggered at Detected Key Point
Recognized Action

Figure 4. Data flow in the sound effect system

The data flow among the main components is illustrated in Figure 4. In the following sections, each component is described in more detail.

3.1 Skeleton Stream

Each frame of video stream contains millions of pixels, which is difficult to classify without dimension reduction. In this report, the video stream is always transformed into skeleton stream using Kinect [4] before being processed. Kinect is a commercially available sensor that leverages stage-of-the-art methods [1] for pose recognition. One key feature of Kinect is to convert video stream into skeleton stream real-timely. Each skeleton frame consists of the coordinates of joints. The preliminary exploration by this report only uses 8 such points: left and right hands, wrists, elbows and shoulders. But the proposed system can be configured to include the other joints with no significant change.

The skeleton stream from Kinect has FPS (frame-per-second) up to 30, which is more than abundant for our application. For better response time, the stream is down sampled to 10 FPS.

Since the skeleton is extracted frame-by-frame, and joint positions in consecutive frames should differ only by relatively small amount, the skeleton of each actor can be tracked in a straight forward manner. Every time a new skeleton is distinguished by the system, a unique label is assigned to it, and in the next frame, the skeleton that is most similar to it will be assigned with the same label. This way, actions performed by different actors will not interfere with each other at all. As it will be shown in the experiment section, number actors on stage does not affect the performance of the proposed system, since the skeleton of each actor is tracked separately.

3.2 Sample Action

The system records every operations by the sound designer and uses it as annotations to the sequence of skeleton frames around the time of the sound effect. The annotated sequence is then used as sample actions to train the gesture recognizer. For each training sample, skeleton frame consists 6 joint points: left (right) hand, wrist and elbow. Each point is a 2D coordinate, where the origin is chosen as the midpoint of the left and right shoulder.

Because only a small range of actions are tested in this report, only 6 joints are used to track the action, and depth information of each joint is ignored. Also we didn’t use the accelerometer of Kinect. These features should be included if the action to be recognized involves legs and 3D movements. In the preliminary exploration by this report, only hand motion is tracked by the 6 joints chosen. But the proposed system can be straight-forwardly extended to support more joints.

In the proposed system, action is represented by a sequence of skeleton frames described above.

3.3 Gesture Recognition with Kernel Nearest-Neighbor

In this report, Kernel Nearest-Neighbor (KNN) algorithm [2] is used to train the recognizer. KNN is a popular method in pattern recognition because of its simplicity and efficiency. The KNN algorithm looks at the k samples in the reference set that are closest to the unknown sample and carries out a vote to make a decision.

As the actor performs, the speed and durations of the movement varies, which bring difficulty to calculate similarity between action samples. Dynamic Time Warping (DTW) [3] is a famous method to compute similarity that is insensitive to time deformations. For this reason, DTW is used as the kernel of KNN to compute similarity between samples.

KNN is a nonparametric method, thus training is not required. The recognition process goes as follows.

(1) Given an input action, use DTW to compute its similarity with each sample actions. Because the input action and sample action might have different length and speed, they are aligned by a "warped" time mapping.
(2) Sort the sample actions by the computed DTW similarity, and select the top k samples. The output is the top action of a vote weighted by similarity among the k sample actions.

3.4 Timing Control

Once an action is recognized, the system need to determine the appropriate timing of playing the sound effect. Different types of actions requires sound effect at different timing. For example, actions involving collision usually generate collision sound at the end of the action, while a wizard casting a spell need gradual sound effect throughout the action. A key point detector is required to control the timing of the sound effect.

In the preliminary exploration by this report, we consider a relatively naive detector that is controlled by a triggering threshold. As the actor is finishing an action, the input action’s similarity with the corresponding samples will gradually increase, thanks to the time-indifferent property of DTW. This monotonicity allows the timing of the sound effect to be controlled by a threshold, which can be learned from the training samples.

4. EXPERIMENT

Since any lag in the triggering of sound effect could bring devastating impact to the performance, the system should be evaluated by at least the following two criteria:

(1) Recognition Criterion. Let the sound designer specifies the actions that need sound effect. The trained action recognition classifier should recognizes these actions with a success rate of almost 100%.
(2) Timing Criterion. Let the sound designer specifies the instant the sound effect should be triggered for the action. The trained key point detector should detect key point that is very close to the designer-specified instant, to the degree that human ear can hardly tell the difference, which can be estimated according to the sensitivity of human ear.

In the rehearsal stage of the experiment, the actor performs a sequence of three kongfu actions (Figure 5): left hand punch, right-hand slap and both-hand dragon shout, during which the sound effect will be added by a third person. In the recognition mode, we first have the original actor performs the full actions twenty times. This way, we can evaluate how well the system meets the recognition criterion by estimating the recognition accuracy. Also a third person is asked to grade the timing of the triggered sound effect, which is used to evaluate the timing criterion. As an extra test, the actor is asked to perform the actions in slow motion, to see if the system generalizes.

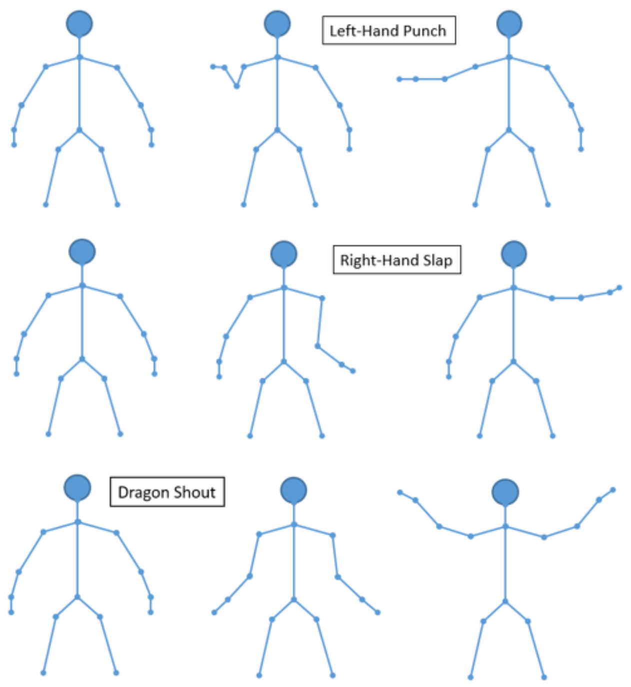

Figure 5. Skeleton for left-hand punch, right-hand slap and both-hand dragon shout.

4.1 Experiment Design

Rehearsal Stage

In rehearsal stage, we have three actors performing on the stage, and one sound designer adding sound effect off the stage. Actor A is the heroine. She is surrounded by bandit actor B and bandit actor C, as shown in Figure 6.

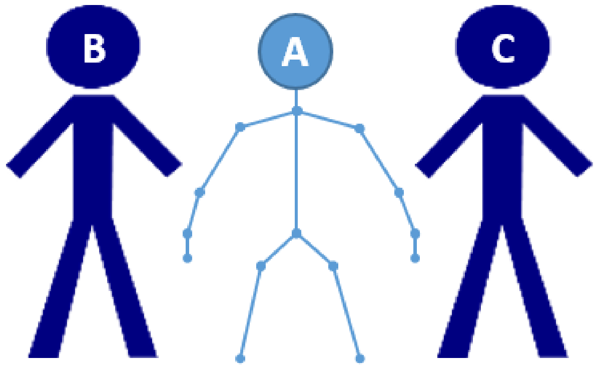

Figure 6. Actor A surrounded by actor B and actor

The punch action should look like actor A hitting actor B at his chest. Though there is no physical contact between the two, actor B should act as if he actually got hit in his chest, and cover the imaginary bruise, as shown in Figure 7. Sound effect for the punch is short and crisp and happens at the end of the stretching of actor A's left arm.

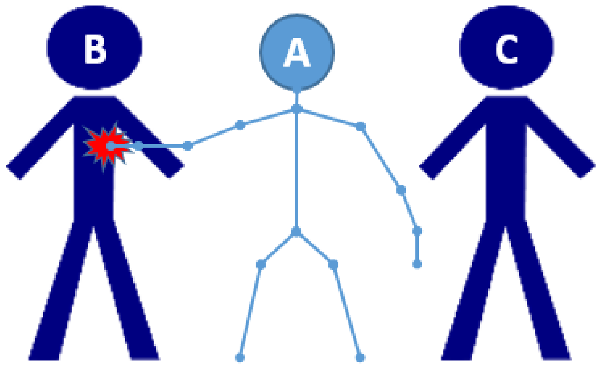

Figure 7. Actor A punching actor B

After actor B get hit and stay motionless, covering his bruise, actor A shall use her right hand to slap actor C, who's trying to attack her from behind. Though the slap is fake, actor C should still act as if he got slapped and move his head towards the other side. This is illustrated in Figure 8. Sound effect for the slap is also short and crisp. It has similar triggering moment with punch.

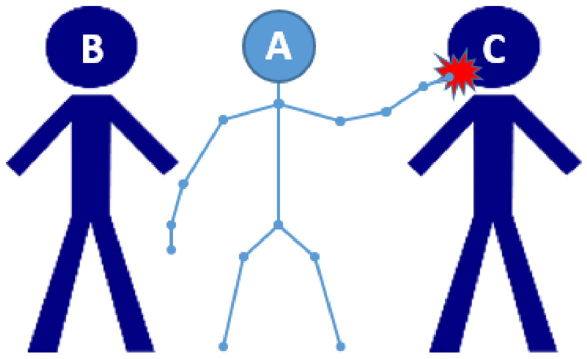

Figure 8. Actor A slapping actor C

Finally after stopping both actor B and actor C's attack, actor A tries to finish them off using dragon shout. As she starts to accumulate energy by raising her both hands, actor B and actor C get scared and tries to run away. This is illustrated in Figure 9. Dragon shout has a gradual sound effect that gets stronger as actor A's hands get higher.

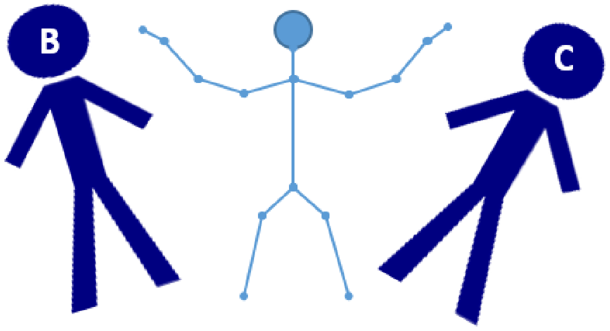

Figure 9. Actor A accumulating energy to use dragon shout

After the above three actions, the recorded actions are used to train the system, so it can be tested in recognition mode.

Recognition Stage

The sound designer now leaves the control room to be sit as an audience, to watch actor A, actor B and actor C repeating the previous performance while grading the timing of the sound effect, since this time, the sound effect is triggered automatically by the trained system. The actors are asked to repeat the acts for large enough number of times, so that both the recognition criterion and timing criterion can be evaluated sufficiently.

Table 1 Test actions and sound effects

ACTION NAME	ACTION TYPE	SOUND EFFECT TYPE
PUNCH	2D	Late
SLAP	3D	Late
DRAGONSHOUT	2D	Gradual

The test actions and sound effects and their types are listed in Table 1. An action is called 3D if significant portion of the action happens on the z-axis of the camera. Late sound effect means the sound effect happens at the tail of the action, while gradual sound effect means the sound effect gradually progresses as the actor carries out the action.

The above actions are performed at the following settings:

- Single actor on stage
- Multiple actor on stage
- Slow motion

Under each setting, the following data is recorded.

Table 2 Result to be recorded

ACTION NAME	RECOGNIZED?	HOW IS THE TIMING
PUNCH	☐ Yes ☐ No	Score (1-10)
SLAP	☐ Yes ☐ No	Score (1-10)
DRAGONSHOUT	☐ Yes ☐ No	Score (1-10)

where the timing is scored by the sound designer.

4.2 Experiment Results

Single Actor on Stage

First, only one actor is on stage, performing three actions in a row. Data listed in Table 2 is recorded which gives the following result in Table 3.

Table 3 Result for Single Actor on Stage

ACTION NAME	RECOGNITION ACCURARCY	TIMING SCORE
PUNCH	100%	10/10
SLAP	95%	10/10
DRAGONSHOUT	100%	10/10

For simple 2D actions (punch and dragon shout) the recognition accuracy is 100% (repeated 20 times). But for 3D action (slap), recognition accuracy is not as perfect.

All three actions are performed by the same actor at the same speed. The reproduced sound effect is perfectly consistent with rehearsal according to the sound designer.

Multiple Actors on Stage

Three actors are on stage. Actor A performs three actions in a row to fight with actor B and C, while they act like getting hit. Same data is recorded as single actor on stage which gives the following result in Table 4.

Table 4 Result for Multiple Actors on Stage

ACTION NAME	RECOGNITION ACCURARCY	TIMING SCORE
PUNCH	100%	10/10
SLAP	95%	10/10
DRAGONSHOUT	100%	10/10

For simple 2D actions (punch and dragon shout) the recognition accuracy is 100% (repeated 20 times). But for 3D action (slap), recognition accuracy is not as perfect.

All three actions are performed by the same actor at the same speed. The reproduced sound effect is perfectly consistent with rehearsal according to the sound designer.

The system's performance is not affected by multiple actors on stage.

Slow Motion
Actor A repeats the actions in extra slow speed, while data listed in Table 2 is recorded which gives the following result in Table 5.

Table 5 Result for Slow Motion

ACTION NAME	RECOGNITION ACCURARCY	TIMING SCORE
PUNCH	100%	10/10
SLAP	100%	9/10
DRAGONSHOUT	100%	7/10

Recognition accuracy is perfect for all three actions.

The sound effect timing for 3D action slap sometimes have noticeable lag. The gradual sound effect for dragon shout is not always in sync with actor's action.

Slow motion improves recognition accuracy for the 3D action slap, but the sound effect timing for slap and dragon shout is compromised.

5. DISCUSSION

From the results in section 4, it can be concluded that the proposed sound effect system:

- Achieves high recognition accuracy in spite of how many actors are on stage or if the action is performed at the same speed.
- Reproduces the timing of the sound effect when actions are being performed at normal speed.

Kinect implements a frame-based skeleton extraction scheme. This allows us to track each individual actor without ambiguity. So multi-actor setting will not affect performance of the system.

Thanks to DTW kernel that is used to recognize actions, the recognizer is insensitive to the speed the actions are being performed, actions performed at a different speed are still recognized correctly. This is very important for stage play, where actors might not be performing actions exactly the same way. In addition, by learning a threshold on the DTW similarity value, the system is able to reproduce the timing of the sound effect.

However, the experiment also suggests that the following problems need to be improved:

- 2D actions can be recognized perfectly. Yet miss happens for 3D action slap, due to lack of 3D information.
- For slow motion, timing control is not as accurate as that of normal speed motion.
- Timing control does not take special care of gradual sound effect, which results in poor performance for slow dragon shout.

Only one Kinect is used in the system and the depth information is ignored. As a result, current system does not perform perfectly for 3D actions. Recognition accuracy can be improved using multiple Kinect sensors installed at different locations.

Also, current threshold-based timing control does not take care of the duration of gradual sound effect, which compromises system's performance in gradual sound effect.

During the experiments, we noticed that for skilled sound designer, the system can get properly trained with as little as just one sample action. For inexperienced sound designer, 3-5 trials might be needed.

6. CONCLUSION

In this report, a gesture recognition system is proposed to trigger sound effects in stage play. The proposed real-time system has three properties: transparency, customizability, and reproducibility. Training samples are automatically collected during rehearsal, where sound designer's operations on the standard sound effect console are recorded as annotations to the happening actions on stage. Once the system gets properly trained, the system is switched to recognition mode, where sound effects will be reproduced by the system and the sound designer no longer need to repeat the same operations.

Promising results are observed in the experiment. The proposed system is accurate and robust even with as little as just one sample action. However, certain problems is also exposed in the experiments.

7. FUTURE WORK

Moving forward, in terms of improving the system, multiple Kinect sensors can be installed on stage to capture full information of 3D actions, so recognition accuracy on 3D actions can be improved.

This report adopts the basic DTW kernel nearest neighbor for gesture recognition. Variations of DTW and other machine learning schemes can be considered in the future to achieve more robust system.

Actions are represented with joint positions in this report. But it has been reported relative angles of body parts could potentially be a better feature set and is more generalizable.

To better reproduce sound designer's timing of the sound effect, more sophisticated timing control scheme is needed to better cope with gradual sound effect and slow motion.

In regards to application, the proposed system can not only be used to add sound effects, but also visual effects and other special effects. Also, the usage of the proposed system may not be limited to stage play, but also special effects in other settings such as video game.

ACKNOWLEDGEMENT
We would like to thank all our classmates who helped us with the experiment and gave us invaluable feedbacks. We thank Prof. Law's insightful comments and suggestions throughout this course. The Kinect sensor used in this report is kindly lent to us by Computational Health Informatics Lab at University of Waterloo.

REFERENCES

1. Shotton, Jamie, et al. "Real-time human pose recognition in parts from single depth images." Communications of the ACM 56.1 (2013): 116-124.
2. Yu, Kai, Liang Ji, and Xuegong Zhang. "Kernel nearest-neighbor algorithm." Neural Process-ing Letters 15.2 (2002): 147-156.
3. Muler, Meinard. "Dynamic time warping." Information retrieval for music and motion (2007): 69-84.
4. Microsoft Corp. Redmond WA. Kinect for Xbox 360.
5. Gavrila, D. M., and L. S. Davis. "Towards 3-d model-based tracking and recognition of human movement: a multi-view approach." International workshop on automatic face-and gesture-recognition. 1995.
6. Suranjith De Silva, Michael Barlow, and Adam Easton. An evaluation of dtw approaches for whole-of-body gesture recognition. In Proceedings of the 28th International BCS Human Computer Interaction Conference on HCI 2014-Sand, Sea and Sky-Holiday HCI, pages 11{21. BCS, 2014.
7. James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. The Journal of Machine Learning Research, 13(1):281{305, 2012.
8. Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In SDM, volume 1, pages 5{7. SIAM, 2001.
9. Ahmad Akl and Shahrokh Valaee. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pages 2270{2273. IEEE, 2010.
10. Camurri, Antonio, et al. "Eyesweb: Toward gesture and affect recognition in interactive dance and music systems." Computer Music Journal 24.1 (2000): 57-69.
11. Qian, Gang, et al. "A gesture-driven multimodal interactive dance system." Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on. Vol. 3. IEEE, 2004.
12. Fails, Jerry Alan, and Dan R. Olsen Jr. "Interactive machine learning." Proceedings of the 8th international conference on Intelligent user interfaces. ACM, 2003.
13. Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.