applied sciences

MDPI

Article

Real-Time Musical Conducting Gesture Recognition Based on a Dynamic Time Warping Classifier Using a Single-Depth Camera

Fahn Chin-Shyurng *, Shih-En Lee and Meng-Luen Wu

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; sammmcakes@gmail.com (S.-E.L.); d10015015@mail.ntust.edu.tw (M.-L.W.)
* Correspondence: csfahn@mail.ntust.edu.tw

Received: 11 October 2018; Accepted: 30 January 2019; Published: 4 February 2019

Abstract: Gesture recognition is a human–computer interaction method, which is widely used for educational, medical, and entertainment purposes. Humans also use gestures to communicate with each other, and musical conducting uses gestures in this way. In musical conducting, conductors wave their hands to control the speed and strength of the music played. However, beginners may have a limited comprehension of the gestures and might not be able to properly follow the ensembles. Therefore, this paper proposes a real-time musical conducting gesture recognition system to help music players improve their performance. We used a single-depth camera to capture image inputs and establish a real-time dynamic gesture recognition system. The Kinect software development kit created a skeleton model by capturing the palm position. Different palm gestures were collected to develop training templates for musical conducting. The dynamic time warping algorithm was applied to recognize the different conducting gestures at various conducting speeds, thereby achieving real-time dynamic musical conducting gesture recognition. In the experiment, we used 5600 examples of three basic types of musical conducting gestures, including seven capturing angles and five performing speeds for evaluation. The experimental result showed that the average accuracy was 89.17% in 30 frames per second.

Keywords: human–computer interaction; dynamic gesture recognition; depth camera; palm tracking; dynamic time warping; musical gesture; musical conductor

1. Introduction

Conducting is the art of directing a musical performance. The primary duties of a conductor include interpreting the scores in a manner that reflects specific indications, keeping the performance on the beat, and communicating with the performers about what emotions are to be presented in the performance. However, to become a professional musical conductor, an individual needs to be equipped with musical knowledge and a sense of musical beat to gesture the tempos correctly. Considerable time is required to become a professional musical conductor. In this study, we developed a musical conducting recognition system to help conducting learners improve their conducting [1]. The recognition system also helps musical performers access the orders with less limitation on playing in terms of weak eye sight and on-going tasks, such as turning a page and looking at the score.

Hand gestures are an important tool that help us communicate with each other. Recently, human–computer interaction, which involves employing new ideas and technologies, has become increasingly important in solving many problems in our daily lives [2]. Gesture recognition involves a human–computer interaction method wherein human gestures are processed and recognized by the machine. Humans often express their actions, intents, and emotions through hand gestures,

and gesture recognition is useful for processing some tasks automatically. For example, American Sign Language helps those with speaking or hearing disabilities to communicate through just hand gestures [3–6]. Another example is traffic police officers who direct traffic during a traffic jam or car accident. Some researchers have also used hand gesture recognition to detect whether a person has Alzheimer's disease [7] or to interpret hand-drawn diagrams [8]. In musical performances, some use the Kinect to recognize drum-hitting gestures [9]. In musical conducting, hand gestures are used to express the conductor's style and emotion and to make performers understand the conductor's guidance [10,11]. Although every conductor has their own style of conducting, a standard conducting rule exists. In 2004, researchers attempted to use computer vision methods to recognize musical conducting gestures [12].

In musical conducting, the conductor is a person who guides the whole performance and keeps everyone on the same tempo. Due to the constant change in tempo and emotion of a musical piece, it is difficult for musicians to constantly focus on the conductor, as they also need to look at the music score and play simultaneously [13]. Musicians may also encounter problems when they have to turn a page while playing or are lost in terms of where they are in the musical piece. It might even be difficult for the musicians to see the conductor if they are sitting far away or have to wear glasses to read the music sheet. To solve these problems, we studied a conductor's gestures and helped the computer recognize and understand the command as the conductor was gesturing. We believe that musicians can use our recognition system to improve their performance during practice and concerts.

Traditionally, the cultivation of a music conductor requires educational training by a mentor so that they can correctly express the timing of a given musical piece [14]. However, this process requires considerable time and the cost of training is high. Therefore, in this study, we developed a system that can help prospective conductors to practice by themselves [15,16]. In musical conducting, there are many types of gestures, including single, double, triple, quadruple, quintuple, sextuple, septuple, and octuple, where the prefixes are taken from the Latin names of the numerals. The gestures are named by their articulation point counts, so there can be infinite types of gestures. In this study, we chose the three most common types of conducting gestures, namely 2/4 duple, 3/4 triple, and 4/4 quad, as shown in Figure 1.

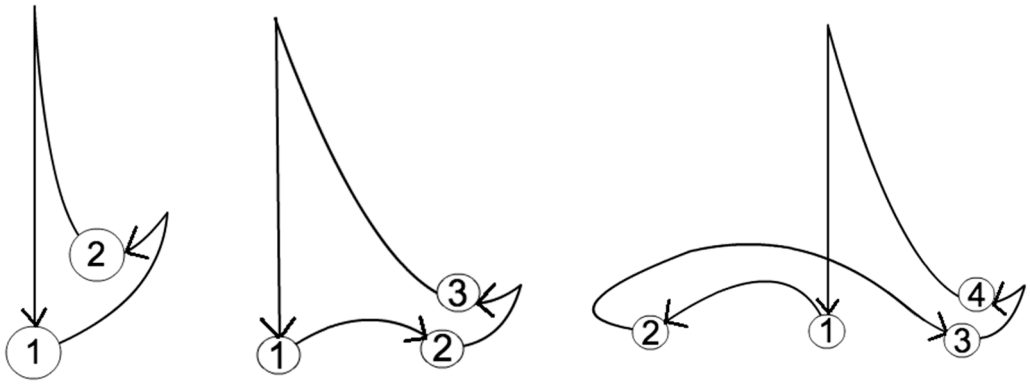


Figure 1. Three basic hand gestures of a musical conductor: (**a**) 2/4 duple, (**b**) 3/4 triple, and (**c**) 4/4 quad.

In computer vision systems, to track the position of hands and palms, a certain amount of preprocessing is required and the accuracy of the performance is not guaranteed. Therefore, in this study, we chose the Kinect to achieve hand tracking [17–20]. The Kinect generates skeleton images of the human body as graphs with edges and vertices through its depth sensors.

We proposed a musical conducting recognition system to classify the types of musical gesture and calculate the conducting speed in order to help conductors improve their performance. In this

section, we have introduced musical conducting gesture recognition and explained how computer vision methods are helpful in assisting musical conductors to perform better. In Section 2, we present our proposed method for musical conducting gesture recognition using the Kinect and dynamic time warping (DTW), in Section 3, the experimental results of our proposed method are presented, and in Section 4, the conclusion and possible future work are described.

2. Proposed Method

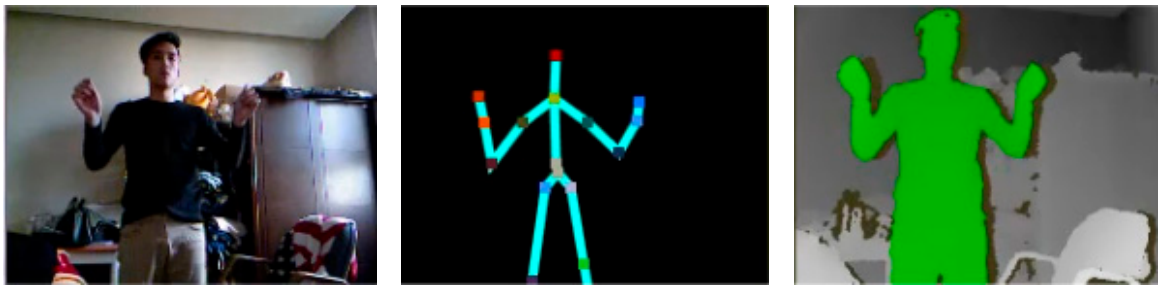
In this section, we describe our proposed method for musical conducting gesture recognition. In the first part, we exploit the Kinect SDK (Software Development Kit) to capture the skeleton image of the conductor, and in the second part, the DTW [21–23] method is used to recognize the conducting gestures. The proposed method can detect multiple gestures at different conducting speeds. A diagram of our proposed method is presented in Figure 2.

Figure 2. Diagram of our proposed method.

Although DTW has been used for various gesture recognitions [24–26], the effectiveness for musical conducting has not been evaluated to a great extent. Our proposed method contributes to musical conducting gesture recognition in an effective manner as such that it is in real time, does not need much computational time, and has a high recognition accuracy.

2.1. Hand Tracking

Our system uses the depth image sensor Xbox 360 Kinect (Microsoft, Redmond, WA, USA, 2013) to capture the required skeleton image of the human body [27], which includes 20 feature points [28]. As in previous literature, for musical gesture recognition, only palm feature points are required [29]. For robustness, there are three inputs in our system, including the original input, a skeleton model, and the depth image, as shown in Figure 3. Among these points, the system uses the significant palm points for musical conducting gesture recognition.



(**a**) (**b**) (**c**)

Figure 3. Three inputs for our proposed system: (**a**) original input, (**b**) skeleton image, and (**c**) depth image with the target body marked in green.

Compared to the methods of extraction and segmentation from the original input, the advantages of using the Kinect for hand tracking is that it is robust to background noise and the colors are similar to the conductor's clothes and skin [30]. Figure 4 shows the example trajectories captured from hand feature points.

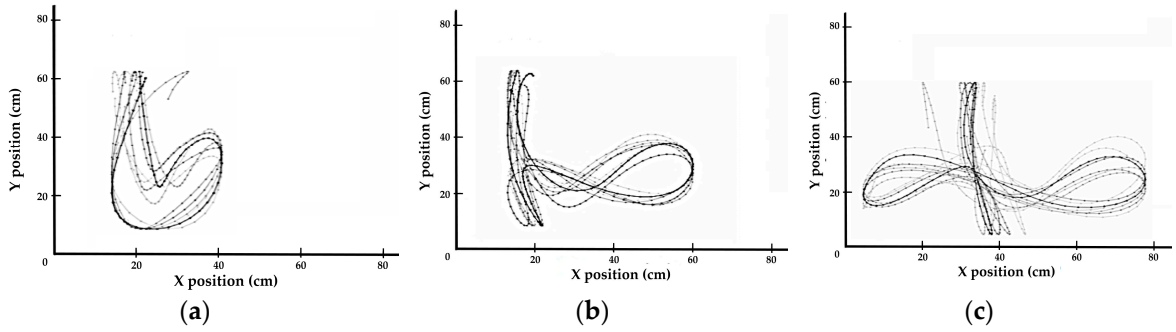


Figure 4. Data points of the continuous gestures captured from tracking users' palms: (**a**) duple, (**b**) triple, and (**c**) quad.

2.2. Separating Continuous Gestures into Single Gestures

In a musical performance, the conducting gesture is continuous, which helps the musicians stay on tempo and prompts them to change the music dynamics. The separation point of two gestures is easy to find for musicians, but not for computer systems. In Figure 5, we show a simplified continuous gesture. The X-axis is the time (t) in seconds, and the Y-axis is the position of the palm of the hand in terms of height (h).

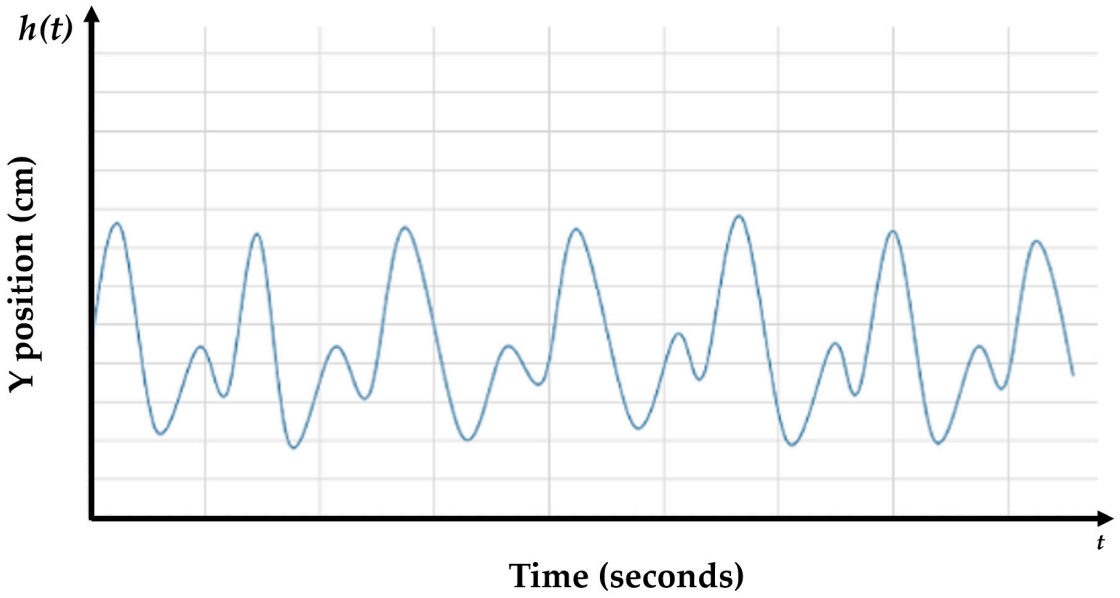


Figure 5. Simplified graph of the height of the palm position during a duple gesture.

In the graph, the palm position is related to its height and the gesture is a duple 2/4 gesture. The height of the starting point is similar to its end point, but dissimilar to other points within the gesture. After successfully segmenting each gesture, the beats per minute can be calculated, and then, the type of gestures can be classified.

To implement gesture segmentation, first, we find the starting position of the whole sequence using the following equation:

$$h(\mathrm{s}) = \inf_{t}(h(t-1)),\ h(t) < h(t-1)\ \&\ h(t-1) \geq h(t-2)\ \&\ t > 2 \quad (1)$$

where $h(s)$ is the starting height of the gesture, $h(t)$ is the current height of the palm, and b is the instance of a beat.

After the first starting point is found, we set the threshold (τ) to the height of $h(s)$. Subsequently, we find the end point of the first gesture (p) by the following equation:

$$h(p) = h(t-1),\ h(t) < h(t-1)\ \&\ h(t-1) \geq h(t-2)\ \&\ h > \tau\ \&\ t > s \quad (2)$$

where τ is a threshold of tolerance, t = $h(s) - \delta$, and δ is set to 15 cm empirically.

Figure 6 is shown as a segmentation result example with their $h(s)$ and $h(p)$s.

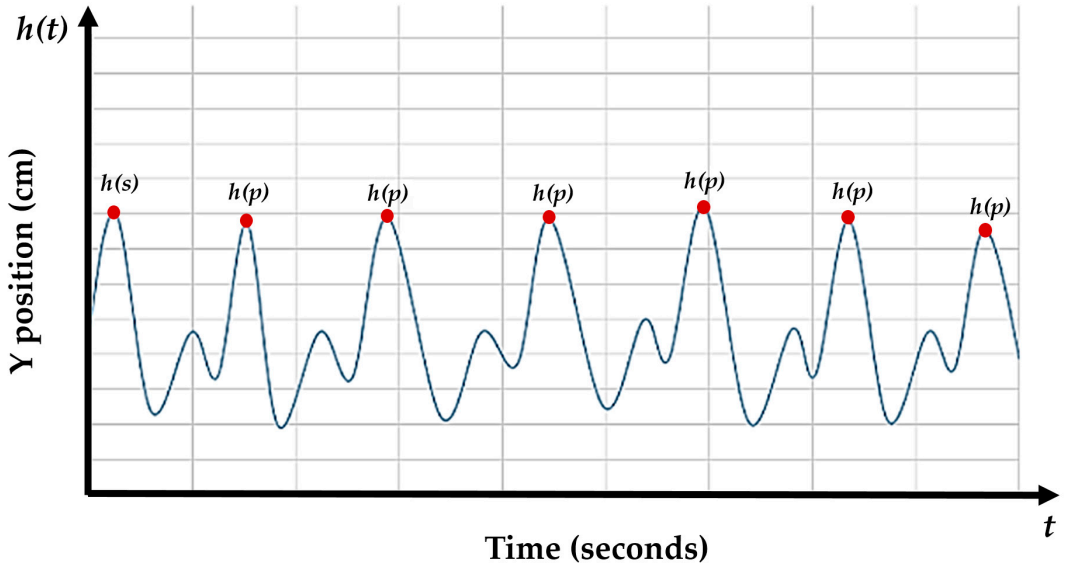


Figure 6. Visualization of a simplified graph of the palm position during a duple gesture after performing segmentation. Figure 6 shows how a continuous gesture is segmented into six different sections, and how each section is individually processed and classified based on the information between the starting and end points. Figure 7 shows one of the segmentations from Figure 6.

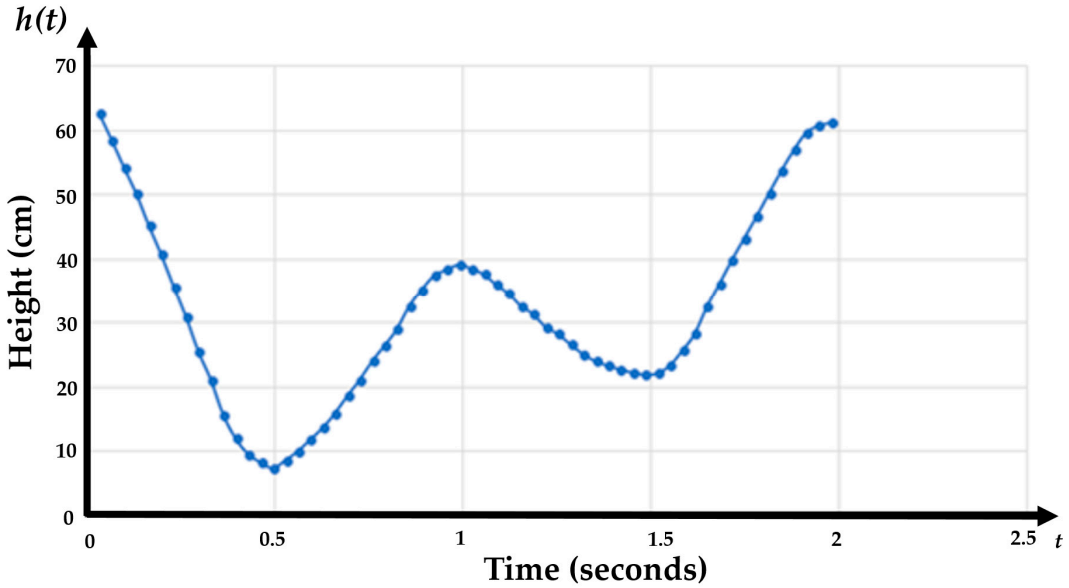


Figure 7. Duple gesture segmentation.

We use relative positions rather than absolute positions in our system, so the method is not affected by the height of the conductor, or the way in which the conductor makes his or her gesture, since the threshold of the height in the algorithm can be changed according to the conductor.

2.3. Tempo Recognition

Depending on the musical score, each measure may or may not have the same tempo, and the conductor is responsible for keeping everyone on the same rhythm. As mentioned earlier, there are three basic gestures that a conductor can make during a performance, but the speed at which each gesture is made is the most important part of keeping the whole orchestra or choir together. The tempo of the beat is universally measured in beats per minute (BPM).

After the segmentation of the gesture in the previous section, we can begin to analyze the data and determine the speed of the gesture. To determine the BPM of the gesture, we first need to know the number of beats made in the gesture and the exact time when the beat was made. We can define this beat as $h(b)$. Once we have this information, we can determine the speed of the gesture and calculate the BPM. Our equation for determining the beats in the gesture is as follows:

$$h(b) = h(t-1),\ h(t) > h(t-1)\ \&\ h(t-1) \leq h(t-2)\ \&\ t > s \tag{3}$$

where b represents the beat, and t and s are the time points with units in seconds. We want to find b that meets the conditions that the height in time, t, is greater than that in time $t-1$, and the height in $t-1$ is also less than that in $t-2$. All ts are greater than the starting time, s.

Once we know where the beats are, we can calculate the BPM of the musical gesture.

After collecting the data for the beats, we can define the BPM of the gesture using the following equation:

$$BPM = \frac{Number\ of\ Beats\ in\ a\ Gesture}{Duration\ of\ the\ Gesture\ (seconds)} \times 60(seconds) \tag{4}$$

The output of the algorithm will give the exact BPM of the gesture—this would work for any gesture irrespective of the number of beats. Capturing the exact tempo will allow the conductor to see if he is making the gesture at the correct speed. The information can also be sent to the musicians if they are looking at a digital music sheet, so they can have a reminder of what speed to play at.

2.4. Dynamic Time Warping (DTW) Classifier

DTW is an algorithm used for measuring similarities between two temporal sequences at different speeds [31]. DTW was originally designed for speech recognition and has been applied to temporal sequences in audio, video, and graphical data [32]. It can be used to analyze any data in a linear sequence. To implement DTW, two time sequences are needed—the original gesture template and the conducting gesture to recognize. DTW takes these two sequences and then calculates an optimal match between the two [33]. The gesture with the best match, in other words, the one with the highest similarity, is the recognized gesture. In this section, we visualize the implementation of DTW. In the visualization, five averaged template samples were used. Figure 8 illustrates the averaged samples in orange and a sample input from the user in blue. A difference was observed between the two, and it was possible to match them correctly by using DTW.

To determine the degree of match, we used a distance matrix (D) to measure the two time sequences. Using the aforementioned example, we took the data and used it to create a distance matrix by employing the DTW algorithm in the following equation:

$$D(h_t(\mathrm{i}),\ h_u(j)) = |h_t(\mathrm{i}) - h_u(j)| + min \begin{pmatrix} D(h_t(i-1), h_u(j)), \\ D(h_t(i-1),\ h_u(j-1)), \\ D(h_t(i),\ h_u(j-1)) \end{pmatrix} \tag{5}$$

where $h_t(i)$ is the time sequence of the known musical gesture of a template, such as the horizontal axis in Figure 9, $h_u(j)$ is the time sequence of an input gesture from the user, such as the vertical axis in

Figure 9, and i and j are the numbers of frames in each sequence. This equation has an initial condition of $D(h_t(1),\ h_u(1)) = |h_t(1) - h_u(1)|$ with the minimum being zero.

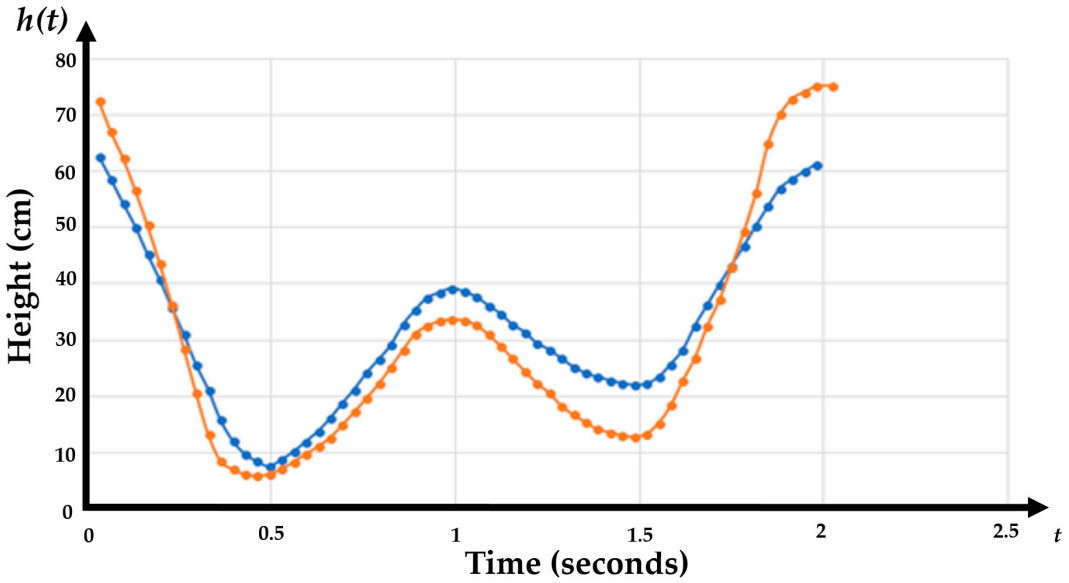


Figure 8. Both the template and user data graphed together, where the blue curve indicates the template gesture and the orange curve indicates the user gesture.

This equation gives a distance matrix between the two time sequences. We can find the warping path between these two sequences by starting from $h_t(1),\ h_u(1)$ and determining the minimum between the distances on the right, below, and diagonally. If the warping path goes diagonally, it is counted as a match. Otherwise, it is just counted as another step in the warping path. To calculate the percentage of accuracy from the template matching, we count the number of matches in the DTW algorithm and compare it with the number of matches that a perfect gesture would have. If the match is over the threshold, the gesture is considered a match. For our system, we set the threshold to 60%. We took the previous example and input the data into the algorithm. The distance matrix and the process of finding an optimal warping path are shown in Figure 9.

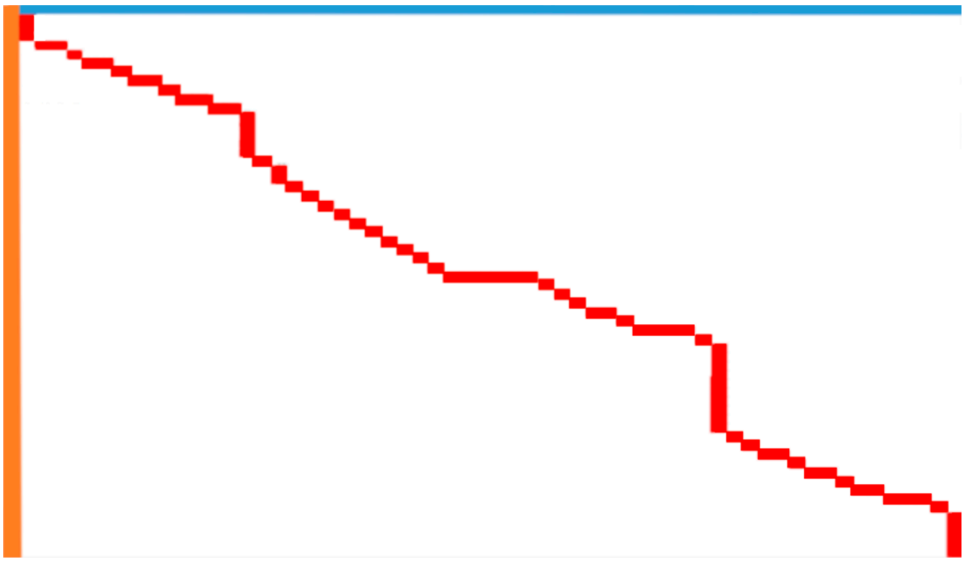
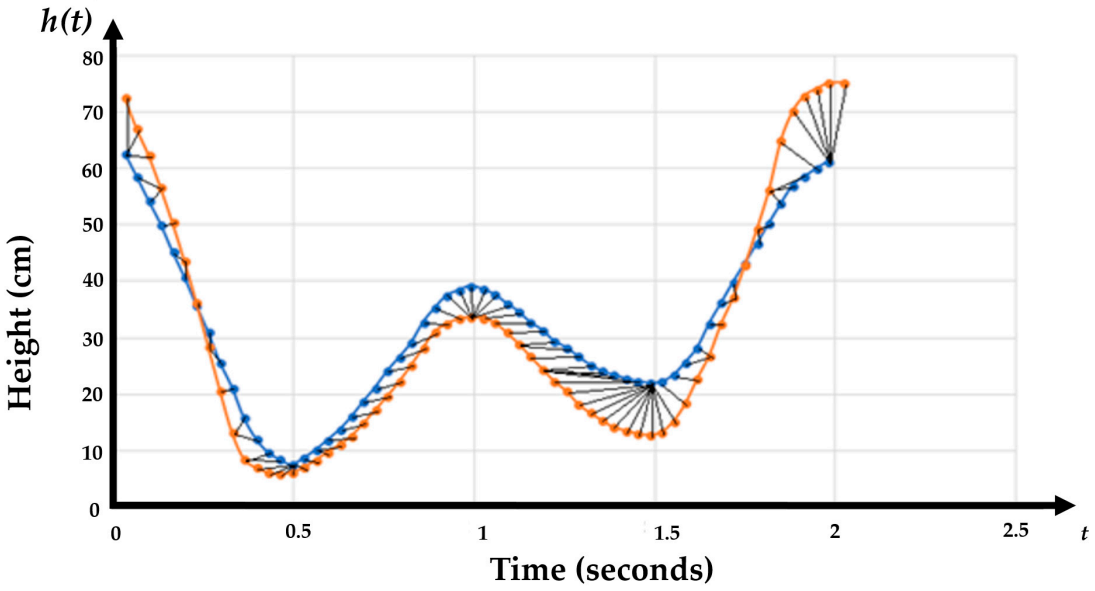


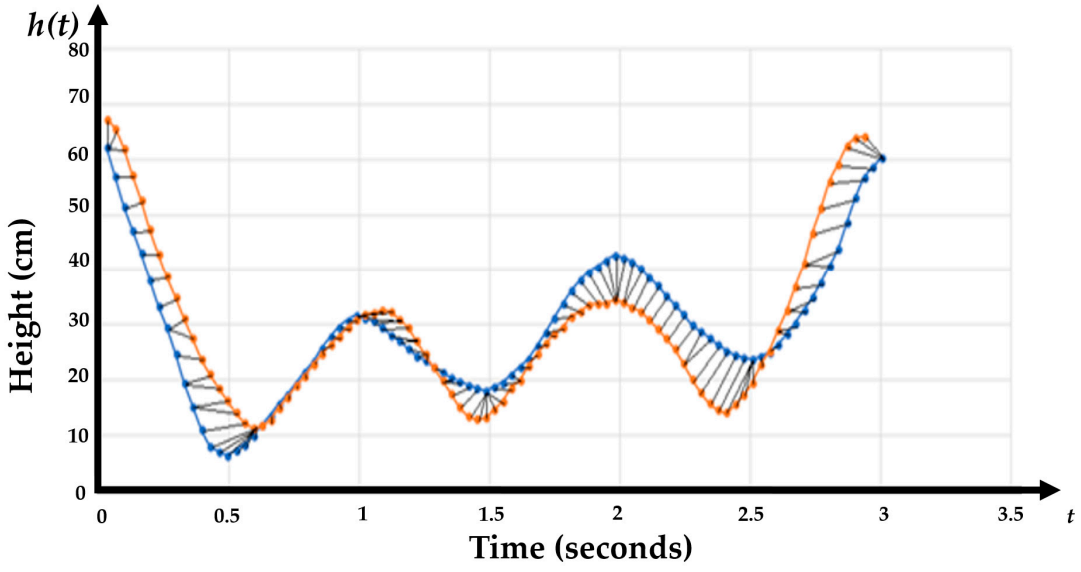
Figure 9. Dynamic warping path marked in red between the template data, marked in blue, and the user data, marked in orange.

This warping path shows where the points of the two gestures are connected and how the path warps according to the distances between the two points. For example, in Figure 9, if there is no point match, the path will move horizontally until the end; otherwise, it will make turns by going down one step. In short, when the path goes down one step, it means that there is a point match. In our system, if the gesture made by the user had more than a 60% match with the template gesture, it was counted as a match and recognized as such. In Figure 10, we visualize the warping path by graphing the points and connecting them to each other where they match.

In this case, the total number of matches in the aforementioned example was 40 out of 60, which provided an accuracy above the threshold. Our system takes each user's gesture and compares it, using DTW, with every template available and records the gesture with the highest accuracy as the output.

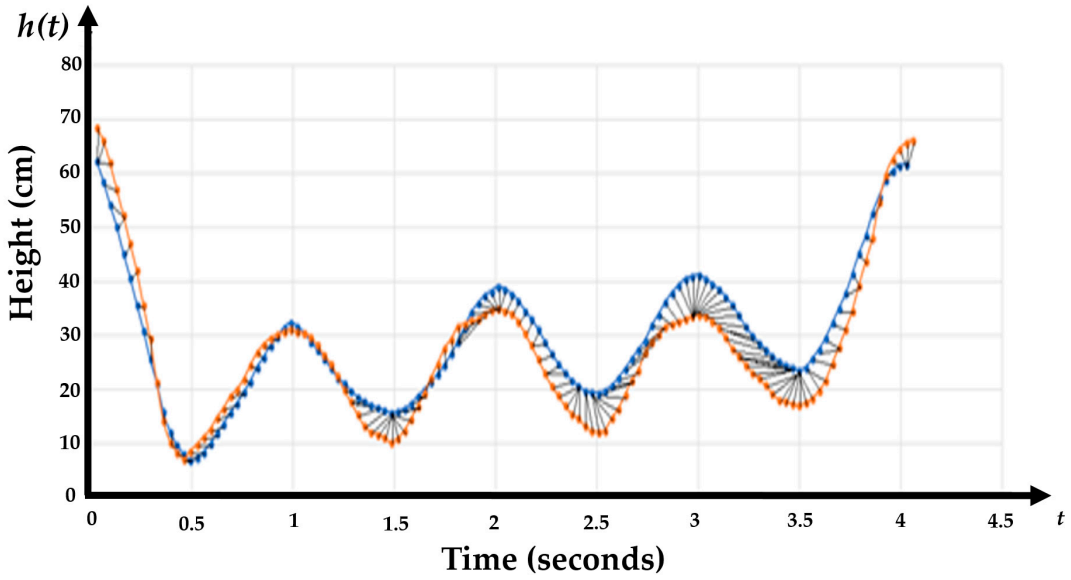


(**a**)



(**b**)

Figure 10. *Cont.*



(**c**)

Figure 10. Visual representation of dynamic time warping (DTW) with the optimal warp, where the blue line represents the template and the orange line represents the users' gesture: (**a**) duple gesture, (**b**) triple gesture, and (**c**) quad gesture.

3. Experimental Results

3.1. Experimental Setup

Our system environment was based on the Microsoft Visual Studio C++ 2010 and Kinect for Windows SDK 1.7. We also used the Xbox 360 Kinect depth camera to film and record our depth videos. The camera filmed at around 30 frames per second and the size of the videos captured was approximately 640 × 640 pixels. Our system was run on a personal computer, and the computer specifications were Intel®Core™ i5-4460 CPU @ 3.20 GHz and 8.00 GB RAM. Table 1 lists the hardware and software devices used in the experiment.

Table 1. Hardware and software used in our system.

Item		Specification
Hardware	Personal Computer	• CPU: Intel®Core™ i5-4460 @ 3.2 GHz • RAM: 8 GB
	Webcam	• Xbox 360 Kinect
Software	Operating System	• Microsoft Windows 7 64-bit
	Developing Tools	• Visual Studio C++ 2010 • Kinect for Windows SDK v1.7

3.2. Database Setup

We developed a database of three gestures made from four different musical teachers and used their gestures to create a template for comparison in our system. In our experiment, the users made the gestures in a continuous motion, one after another, and the algorithm recognized the gesture as it was being made. A total of 5600 gestures were made. There were 1750 gestures for each basic music gesture and 350 for other gestures that may appear in a performance but are not tempo gestures, such as turning the page or pointing at a section of the orchestra. These gestures were made in a combination of different BPM and facing the camera at different angles. We also gestured to five different BPM for the same interval of time to find the mean squared error and to show the robustness of our system. Finally, we gestured along to three complete songs. Our experimental results are separated into two

parts—the accuracy of the recognition of single musical gestures and the accuracy of the recognition of continuous musical gestures. All our experimental videos were filmed indoors. In Figures 11–13, we show the three basic gestures being made at different speeds.

Figure 11. Two examples of a duple (2/4) gesture.

Figure 12. Examples of a triple (3/4) gesture.

Figure 13. Examples of a quad (4/4) gesture.

3.3. Results of Single Musical Gesture Recognition

In our experiment, we used the template shown in Section 3.2 to test and verify the gesture recognition results. Five people used the real-time system to perform the dynamic gesture recognition.

The users had to stand in front of the camera with nothing in between so that the camera could accurately detect where the palms were. The background of the footage and the clothing of the user did not affect the results, because we used a depth camera. Only one user's gestures can be detected at a time, and our system detects each of the user's palms separately. To verify the result, we used half of the data to train the model and another half to verify the accuracy of the model. The accuracy of the musical gesture recognition can be defined as follows [34]:

$$Accuracy = \frac{number\ of\ correctly\ classified\ data}{number\ of\ testing\ data} \tag{6}$$

The original images, skeleton images, and the depth maps of the frames of the duple, triple, and quad musical conducting gestures are shown in Figures 14–16, respectively. Each user performed 350 gestures for each musical gesture and 70 gestures for the other category. The accuracy of the experiment is presented in Table 2.

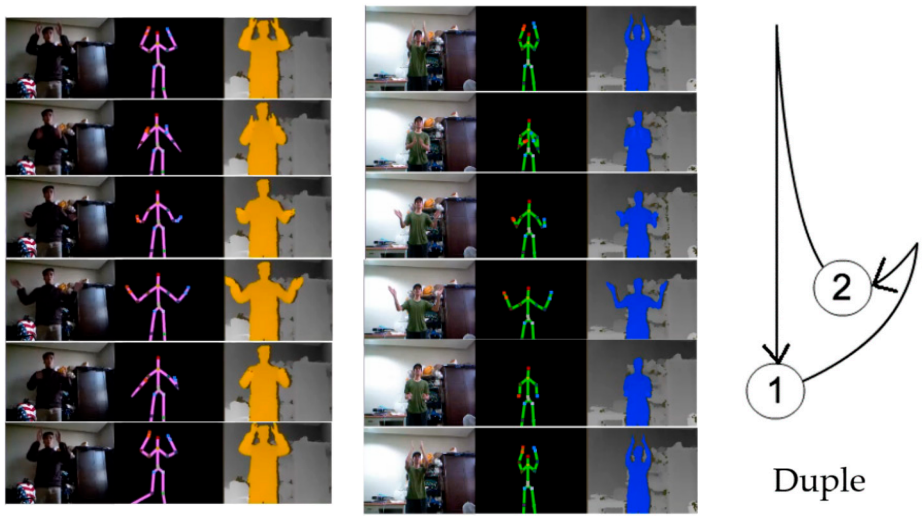


Figure 14. Screenshots of a single musical conducting gesture (duple) being made while facing the camera, including the 2D video, a skeleton model of the musical conductor, and the depth video.

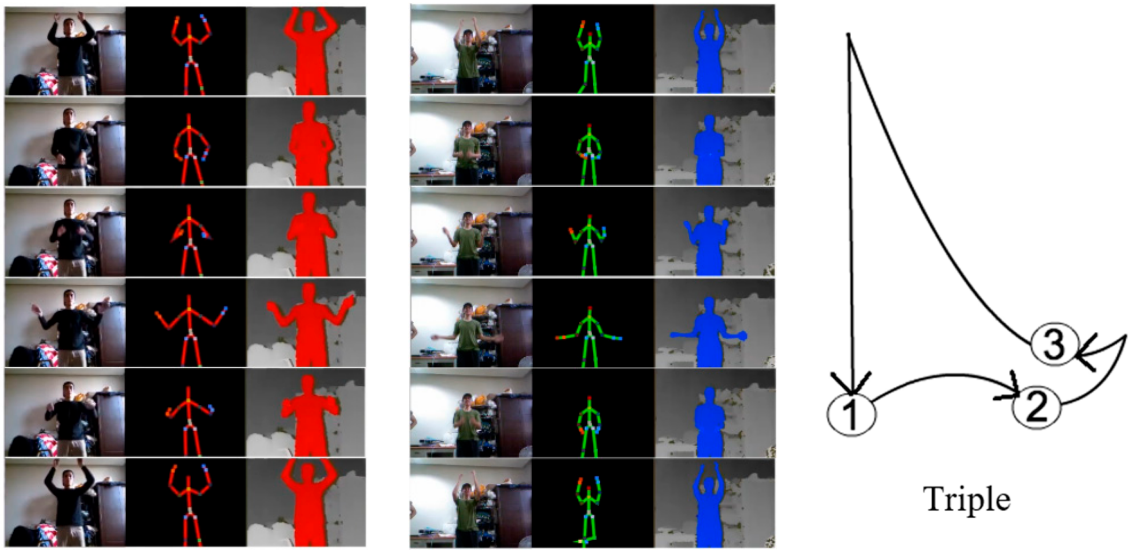


Figure 15. Screenshots of a single musical conducting gesture (triple) being made while facing the camera, including the 2D video, a skeleton model of the musical conductor, and the depth video.

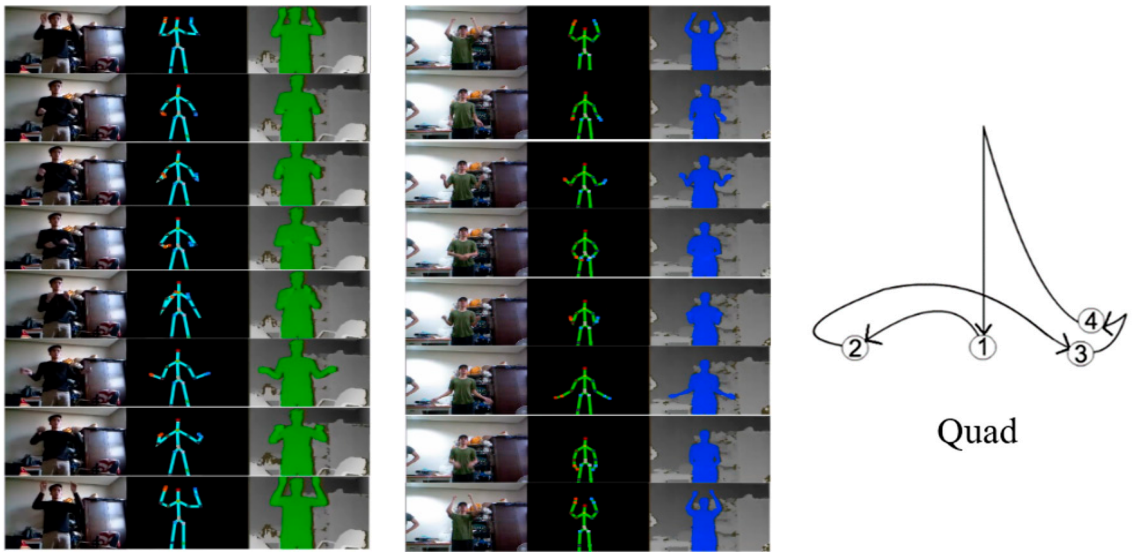


Figure 16. Screenshots of a single musical conducting gesture (quad) being made while facing the camera, including the 2D video, a skeleton model of the musical conductor, and the depth video.

Table 2. Accuracy of musical conducting gesture recognition.

Musical Gesture	Number of Testing Data	Number of Correctly Classified Gestures	Number of Misclassified Gestures	Accuracy
Duple	1750	1633	117	93.31%
Triple	1750	1571	179	89.77%
Quad	1750	1458	292	83.31%

Conducting gestures at different speeds affects the accuracy. Table 3 shows the conducting gesture recognition result of different conducting speeds. In the experiment, we used a metronome to assist the user in staying at a constant BPM. The metronome allowed us to determine the exact tempo that the user should be gesturing at, which made the experiment more stable. There were still some misclassifications in our experiment, and one of the main reasons for this is that the speed was too high for the camera to capture the gesture clearly. Our camera could only capture 30 frames per second, which made it difficult to capture a gesture made at 200 BPM, which is approximately 3.33 beats a second. In addition, because the gesture was made so quickly, the movement was much slower to compensate for the quickness, which made it difficult for our system to clearly capture the hand gesture.

Table 3. Accuracy of musical conducting gesture recognition classified by BPM.

Musical Gestures	Number of Testing Data	Number of Correctly Classified Data	Number of Misclassified Data	Accuracy
Duple (150 BPM)	350	334	16	95.43%
Triple (150 BPM)	350	319	31	91.14%
Quad (150 BPM)	350	295	55	84.29%
Duple (200 BPM)	350	249	101	71.14%
Triple (200 BPM)	350	202	148	57.71%
Quad (200 BPM)	350	113	237	32.29%
Other	350	316	34	90.29%

As seen in Table 4, the accuracy of the classification below 150 BPM was 100% and thus ideal for being used in the real world. However, as the speed reached 150 BPM and higher, the accuracy dropped. The total accuracy at each speed is shown in Table 4.

Table 4. Accuracy of musical gestures classified by BPM.

Tempo	Number of Testing Data	Number of Correctly Classified Data	Number of Misclassified Data	Accuracy
60 BPM	1050	1050	0	100%
90 BPM	1050	1050	0	100%
120 BPM	1050	1050	0	100%
150 BPM	1050	948	102	90.29%
200 BPM	1050	564	486	53.71%

As seen in the above table, the accuracy at 150 BPM started to drop, and at 200 BPM, the accuracy dropped dramatically. As mentioned before, the main reason for the drop in accuracy was that the camera filmed at 30 frames per second, and the resolution of a gesture decreased as the BPM increased. Fortunately, in real-life applications, there are very few musical pieces that actually go up to 200 BPM, and thus, our proposed system is usable on most occasions.

Our experiment comprised conducting gesture recognition not only at different speeds but also at different angles. Our system can accurately recognize the gesture as long as the palms are clearly captured by the depth camera, because it recognizes the gestures based on the vertical movement of the palms. The accuracy of our musical conducting recognition system at different viewing angles is shown in Table 5.

Table 5. Accuracy of musical conducting gesture recognition at different viewing angles.

Angles	Number of Gestures Made	Number of Correctly Classified Data	Number of Misclassified Data	Accuracy
45° Upward	800	713	87	89.13%
45° Downward	800	716	84	89.50%
0° Facing Camera	800	724	76	90.50%
−45° Left	800	715	85	89.38%
−90° Left	800	702	98	87.75%
45° Right	800	710	90	88.75%
90° Right	800	698	102	87.25%

At 90°, a part of the conductor's body was blocked, so the accuracy of the gesture was slightly lower. Our method can recognize three basic gestures made at 60−200 BPM and can recognize the gestures at any angle within 45° upward or downward and within 90° to the left or right.

3.4. Results of Continuous Musical Gesture Recognition

In previous experiments, each gesture was separated from each other and only one gesture was input at a time. However, in this experiment, we used actual gestures in a performance where all the gestures were connected and continuous. For the continuous gestures, the experiment was divided into two parts. In the first part, we used continuous gestures at five different BPM and calculated the mean squared error between the computed BPM and the actual ones. Table 6 lists the results of the average squared errors in the gestures of different BPM.

Table 6. Mean squared error results.

Speed of the Gesture	Number of Gestures	Mean Squared Error
60 BPM	75	0.3 BPM
90 BPM	97	0.4 BPM
120 BPM	136	0.6 BPM
150 BPM	173	0.9 BPM

The mean squared error shows the average error that the user makes when making the gestures.

In the second part of our experiment, we chose three musical pieces for the actual performance. Unlike the previous experiments, in this part of the experiment, the musical conductors changed their gesture styles in different parts of the music to convey its emotion. Table 7 shows the results of musical gesturing along with the music.

Table 7. Accuracy of musical conducting gesture recognition in a whole musical piece.

Musical Piece	Number of Gestures Made	Number of Correctly Classified Data	Number of Misclassified Data	Accuracy
Canon in D	94	94	0	100%
Mozart Symphony No. 40	107	107	0	100%
Hallelujah	111	110	1	99.10%

The experimental result shows that our system can recognize the gestures accurately with variable BPM.

3.5. Comparison of Existing Methods and Our Method

As seen from our experimental results, we achieved a total accuracy of 89.17%. Our experiment comprised five tempo ranges, three gestures, and seven filming angles. Our method can also be applied to a complete song with very few errors.

Our experimentation is relatively more complete compared with similar methods of recognizing a musical gesture. Being a musical conductor means moving around and turning toward different musicians at different times, and our method allows a range of 180°. It also allows the camera to be above or below the conductor, which is more useful in the real world. For example, if a camera is placed directly in front of the conductor, it may block some musicians and will not be aesthetically pleasing for the audience. However, if a camera is hidden behind the musicians or above the conductor, it will not block the musicians or the conductor and the audience will not be able to see it.

We compared our method with the methods reported by three other studies that are closely related to ours. The first one is by H. Je, J. Kim, and D. Kim, who used a depth camera to recognize some musical gestures through feature points [5]. The second one is by N. Kawarazaki, Y. Kaneishi, N. Saito, and T. Asakawa [35], who used a depth camera and an inertia sensor to detect the conductor's movements and where the conductor was facing. The final method is by S. Cosentino, Y. Sugita, and M. Zecca [36], who used an inertia glove that could transmit signals to a robot to let the robot know what gesture the robot was making. Tables 8 and 9 show the comparisons of the related work with our method.

Table 8. Comparisons of related work with our method.

Experimentations	H. Je et al.	N. Kawarazaki et al.	S. Cosentino et al.	Our Method
BPM 50-60	✔	✔	✔	✔
BPM 61-90	✔	✔	✔	✔
BPM 91-120	✔	✔	✔	✔
BPM 121-150				✔
BPM 150-200				✔
Duple (2/4)	✔	✔		✔
Triple (3/4)	✔	✔		✔
Quad (4/4)	✔	✔	✔	✔
Facing Camera	✔	✔	✔	✔
−45° Left		✔		✔
−90° Left		✔		✔
45° Right		✔		✔
90° Right		✔		✔
45° Upwards				✔
90° Downwards				✔
Depth Camera	✔	✔		✔
Inertia Sensors		✔	✔	

Table 9. Accuracy comparison of related work with our method.

Method	H. Je et al.	N. Kawarazaki et al.	S. Cosentino et al.	Ours
Accuracy	79.75%	78.00%	90.10%	89.17%

Although our method did not reach the highest accuracy when compared with the other methods, we experimented with more gestures, angles, and tempos than the method with the highest accuracy. In addition, we only used a depth camera for our experiments, which is cost-effective for the results we achieved.

4. Conclusions

This paper proposes a musical conducting recognition system that can classify three basic musical conducting gestures and recognize these gestures up to 150 BPM. We used the Kinect to capture the depth map of a musical conductor and formed a skeleton model with palm data points. The time-varying palm data points were formed as a sequence of our input. The value of the data points was relative to the musical conductor's height, so the system was adaptive to the height of most users. Our proposed method is based on a DTW algorithm, which is a dynamic programming method for sequential data analysis. Combined with the Kinect depth camera and the Kinect SDK, the system can capture and track the movement of users' palms and classify the gestures. Before classification, the continuous gestures are segmented into separate ones. Additionally, the BPM of each gesture is calculated to help users practice the gestures. Once the BPM of the gesture is determined, the gesture can be identified and classified. We collected the musical gestures of three musical professionals and combined their gestures to create a template for comparison in our DTW algorithm.

In the experiment, we used three different musical gestures of five users at five different speeds and seven different angles for evaluation. We collected all the results and achieved a total accuracy of 89.17%. Our system can recognize three different musical gestures at five different speeds and seven different angles, with the fastest speed being 200 BPM and the largest angle being 90°, with an overall frame rate of 30 frames per second (FPS).

In this study, we recognized the three most common gestures in musical conducting, which are the "duple", "triple", and "quadratic". For future work, we recommend that more types of musical gestures should be added for recognition. Also, the current tolerance angle between the musical conductor and the camera is 90% from left or right and 45% from upper or lower, which can be extended. Finally, in this experiment, the recognition often failed when a gesture was halted abruptly, which means that the robustness of the system to a sudden unexpected change in the behavior of musical conductors can be improved.

Author Contributions: The authors contribute equally to this work.

Funding: The authors thank the Ministry of Science and Technology, Taiwan (R.O.C.), for supporting this work in part (MOST-107-2221-E-011-113-MY2).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Forrester, H.S. Music teacher knowledge: An examination of the intersections between instrumental music teaching and conducting. *J. Res. Music Educ.* **2018**, *65*, 461–482. [CrossRef]
2. Cavalieri, L.; Mengoni, M.; Ceccacci, S.; Germani, M. A Methodology to Introduce Gesture-Based Interaction into Existing Consumer Product. In Proceedings of the International Conference on Human-Computer Interaction, Toronto, ON, Canada, 17–22 July 2016; pp. 25–36. [CrossRef]

3. Ahmed, W.; Chanda, K.; Mitra, S. Vision based hand gesture recognition using dynamic time warping for Indian sign language. In Proceedings of the International Conference on Information Science, Kochi, India, 12–13 August 2016; pp. 120–125. [CrossRef]
4. Lichtenauer, J.F.; Hendriks, E.A.; Reinders, M.J. Sign language recognition by combining statistical DTW and independent classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 2040–2046. [CrossRef] [PubMed]
5. Klomsae, A.; Auephanwiriyakul, S.; Theera-Umpon, N. A novel string grammar unsupervised possibilistic C-medians algorithm for sign language translation systems. *Symmetry* **2017**, *9*, 321. [CrossRef]
6. Galka, J.; Masior, M.; Zaborski, M.; Barczewska, K. Inertial motion sensing glove for sign language gesture acquisition and recognition. *IEEE Sens.* **2016**, *16*, 6310–6316. [CrossRef]
7. Negin, F.; Rodriguez, P.; Koperski, M.; Kerboua, A.; Gonzàlez, J.; Bourgeois, J.; Chapoulie, E.; Robert, P.; Bremond, F. PRAXIS: Towards automatic cognitive assessment using gesture recognition. *Expert Syst. Appl.* **2018**, *106*, 21–35. [CrossRef]
8. Costagliola, G.; Vincenzo, V.; Risi, M. A Multi-layer Parsing Strategy for On-line Recognition of Hand-drawn Diagrams. In Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing, Brighton, UK, 4–8 September 2006; pp. 103–110. [CrossRef]
9. Rosa-Pujazón, A.; Barbancho, I.; Tardón, L.J.; Barbancho, A.M. Fast-gesture recognition and classification using Kinect: An application for a virtual reality drumkit. *Multimed. Tools Appl.* **2016**, *75*, 8137–8164. [CrossRef]
10. Je, H.; Kim, J.; Kim, D. Vision-based hand gesture recognition for understanding musical time pattern and tempo. In Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei, Taiwan, 5–8 November 2007; pp. 2371–2376. [CrossRef]
11. Fabiani, M.; Friberg, A.; Bresin, R. Systems for interactive control of computer generated music performance. In *Guide to Computing for Expressive Music Performance*; Springer: London, UK, 2013; pp. 49–73.
12. Kolesnik, P. Conducting Gesture Recognition, Analysis and Performance System. Master's Thesis, McGill University, Montreal, QC, Canada, 2004.
13. Chen, S.; Maeda, Y.; Takahashi, Y. Melody oriented interactive chaotic sound generation system using music conductor gesture. In Proceedings of the IEEE International Conference on Fuzzy Systems, Beijing, China, 6–11 July 2014; pp. 1287–1290. [CrossRef]
14. Toh, L.W.; Chao, W.; Chen, Y.S. An interactive conducting system using Kinect. In Proceedings of the IEEE International Conference on Multimedia and Expo, San Jose, CA, USA, 15–19 July 2013; pp. 1–6. [CrossRef]
15. Nijholt, A.; Reidsma, D.; Ebbers, R.; Maat, M. The virtual conductor-learning and teaching about music, performing, and conducting. In Proceedings of the IEEE International Conference on Advanced Learning Technologies, Santander, Cantabria, Spain, 1–5 July 2008; pp. 897–899. [CrossRef]
16. Fazekas, G.; Barthet, M.; Sandler, M.B. Mood conductor: Emotion-driven interactive music performance. In Proceedings of the Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, Switzerland, 2–5 September 2013; p. 726. [CrossRef]
17. Ren, Z.; Meng, J.; Yuan, J.; Zhang, Z. Robust part-based hand gesture recognition using Kinect sensor. *IEEE Trans. Multimed.* **2013**, *15*, 1110–1120. [CrossRef]
18. Yavsan, E.; Ucar, A. Gesture imitation and recognition using Kinect sensor and extreme learning machines. *Measurement* **2016**, *94*, 852–861. [CrossRef]
19. Hachaj, T.; Ogiela, M.R.; Koptyra, K. Human actions recognition from motion capture recordings using signal resampling and pattern recognition methods. *Ann. Oper. Res.* **2018**, *265*, 223–239. [CrossRef]
20. Xi, C.; Zhao, C.; Pei, Q.; Liu, L. Real-time Hand Tracking Using Kinect. In Proceedings of the International Conference on Digital Signal Processing, Shanghai, China, 19–21 November 2018; pp. 37–42. [CrossRef]
21. Kshirsagar, M.S.; Annadate, M.N. Survey on Music Conducting Gestures using Dynamic Time Warping. *Int. Res. J. Eng. Technol.* **2017**, *4*, 2835–2839. [CrossRef]
22. Raheja, J.L.; Minhas, M.; Prashanth, D.; Shah, T.; Chaudhary, A. Robust gesture recognition using Kinect: A comparision between DTW and HMM. *Int. J. Light Electron Opt.* **2015**, *126*, 1098–1104. [CrossRef]
23. Sahoo, J.P.; Ari, S.; Ghosh, D.K. Hand gesture recognition using DWT and F-ratio based feature descriptor. *IET Image Process.* **2018**, *12*, 1780–1787. [CrossRef]
24. Tseng, M.; Korolik, V.; Scherer, S.; Matarić, M. Comparing models for gesture recognition of children's bullying behaviors. In Proceedings of the International Conference on Affective Computing and Intelligent Interaction, San Antonio, TX, USA, 23–26 October 2017; pp. 138–145.

25. Liu, Y.T.; Zhang, Y.A.; Zeng, M. Novel Algorithm for Hand Gesture Recognition Utilizing a Wrist-Worn Inertial Sensor. *IEEE Sens. J.* **2018**, *18*, 10085–10095. [CrossRef]
26. Kwon, M.C.; Park, G.; Choi, S. Smartwatch User Interface Implementation Using CNN-Based Gesture Pattern Recognition. *Sensors* **2018**, *18*, 2997. [CrossRef] [PubMed]
27. Pal, D.H.; Kakade, S.M. Dynamic hand gesture recognition using Kinect sensor. In Proceedings of the International Conference of Global Trends in Signal Processing, Information Computing and Communication, Jalgaon, India, 22–24 December 2016; pp. 448–453. [CrossRef]
28. Chavarria, H.V.; Escalante, H.J.; Sucar, L.E. Simultaneous segmentation and recognition of hand gestures for human-robot interaction. In Proceedings of the 16th International Conference on Advance Robotics, Montevideo, Uruguay, 25–29 November 2013; pp. 1–6. [CrossRef]
29. Bradshaw, D.; Ng, K. Tracking conductors hand movements using multiple wiimotes. In Proceedings of the International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution, Florence, Italy, 17–19 November 2008; pp. 93–99. [CrossRef]
30. Zhang, Q.Y.; Zhang, M.Y.; Hu, J.Q. A method of hand gesture segmentation and tracking with appearance based on probability model. In Proceedings of the 2nd International Symposium on Intelligent Information Technology Application, Shanghai, China, 20–22 December 2008; pp. 380–383. [CrossRef]
31. Hsu, C.J.; Huang, K.S.; Yang, C.B.; Guo, Y.P. Flexible Dynamic Time Warping for Time Series Classification. In Proceedings of the International Conference on Computer Science, Reykjavík, Iceland, 1 January 2015; pp. 2838–2842. [CrossRef]
32. Nigam, S.; Singh, R.; Misra, A.K. A Review of Computational Approaches for Human Behavior Detection. *Arch. Comput. Methods Eng.* **2018**, 1–33. [CrossRef]
33. Plouffe, G.; Cretu, A. Static and dynamic hand gesture recognition in depth data using dynamic time warping. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 305–316. [CrossRef]
34. Glowacz, A.; Glowacz, Z. Recognition of images of finger skin with application of histogram, image filtration and K-NN classifier. *Biocybern. Biomed. Eng.* **2016**, *36*, 95–101. [CrossRef]
35. Kawarazaki, N.; Kaneishi, Y.; Saito, N.; Asakawa, T. A supporting system of chorus singing for visually impaired persons using depth image sensor. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Kaohsiung, Taiwan, 11–14 December 2012; pp. 1–4. [CrossRef]
36. Cosentino, S.; Sugita, Y.; Zecca, M. Music conductor gesture recognition by using inertial measurement system for human-robot musical interaction. In Proceedings of the IEEE International Conference on Robotics and Biomimetrics, Guangzhou, China, 11–14 December 2012; pp. 30–35. [CrossRef]