ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Dancing Scoring Robot

Team #29

XIAOHAN ZHU
(xzhu54@illinois.edu)
HEYUE WANG (heyuew2)
CHUXUAN HU
(chuxuan3@illinois.edu)
YIYUAN CHEN
(yiyuanc2@illinois.edu)

TA: Tielong Cai

May 18, 2023

Abstract

Nowadays, people lack convenient and objective means to evaluate their dances in everyday entertainment, and existing dance-scoring products like JustDance on Xbox necessitate complex equipment but only offer relatively simplistic evaluation. Simultaneously, the demand for diverse dance style evaluations is growing, which has motivated us to develop a user-defined dance scoring system. Through the culmination of our senior project development efforts, we have successfully created an easily accessible and affordable PC-based dance scoring machine. This system also incorporates a team-developed bracelet, enabling evaluation across multiple aspects and flexible assessment criteria.

Contents

1 Introduction

1.1 Purpose: Problem and solution

The problem that the dancing scoring robot addresses is that people who engage in dance for personal entertainment and exercise often lack access to expert feedback on their technique and performance quality. In traditional dance settings, such as dance studios or fitness classes, instructors may not have the time or resources to provide individualized feedback to every participant. This can lead to frustration and a lack of motivation to continue dancing.

The dancing scoring robot provides a solution to this problem by offering personalized evaluations of the user's dance performance, including feedback on elements such as rhythm, timing, and posture. By using the machine, dancers can receive immediate feedback on their performance, allowing them to make adjustments and improve their skills in real time. Additionally, the machine's scoring system can provide a fun and engaging way for users to track their progress and challenge themselves to improve their scores. The dancing scoring robot thus provides an accessible and effective means for people to improve their dance skills and achieve their fitness goals.

Although there are existing products with similar goals, these products evaluate users based on simple and, in most cases, mono criteria. For example, Just Dance using X-box[1] evaluates users' performance based merely on their pose matches, neglecting the continuity and fluency of their motions. Our team aims to resolve these limitations by devising a solution that achieves more objective dance scoring by implementing three distinct evaluation methods. We will utilize hardware components such as a camera, smart bracelet, processor, storage, and display to create a robot capable of scoring the dancers' performance. The robot will have good human-computer interaction to enhance user experience.

To ensure a comprehensive evaluation of the dancers' performance, we will adopt three different methods for evaluation. Firstly, we will evaluate whether the dancer's movements are based on both pose matches and motion continuity. Secondly, we will assess how well the dancer's movements match the dance music. Lastly, we will evaluate the dancer's body condition in real-time, analyze the intensity of their movements, and record the dancer's hand movements in greater detail. By integrating these three evaluation methods, we can create a robust and comprehensive evaluation of the dancer's performance, which will be displayed on the screen. This solution will enable judges to make more objective decisions and provide dancers with valuable feedback to improve their performance.

1.2 High-Level Requirement List

- The dancing scoring robot contains the bracelet and the PC, which is portable and easily accessed by users.

[1] https://www.xbox.com/en-US/games/store/just-dance-2022/9N04KQK2LBZL

- The dancing scoring robot should be able to generate multiple aspects of evaluations on users' dances, including motion matching, rhythm matching, and exercise effect.
- The user could define the 'standards' of the evaluation by uploading the reference video. Then the scores would be given out with respect to the user-defined 'standards'.
- The dancing scoring robot should be user-friendly and interactive with our human-computer interaction subsystem. All subsystem evaluation results are clearly labeled on our UI.
- The dancing scoring robot should be efficient. A 15s long dancing video could be processed and evaluated within 5 minutes.
- The evaluations generated by the dancing scoring robot are reasonable and aligned with intuitions, which have been verified by several users' feedback.

1.3 Block Diagram

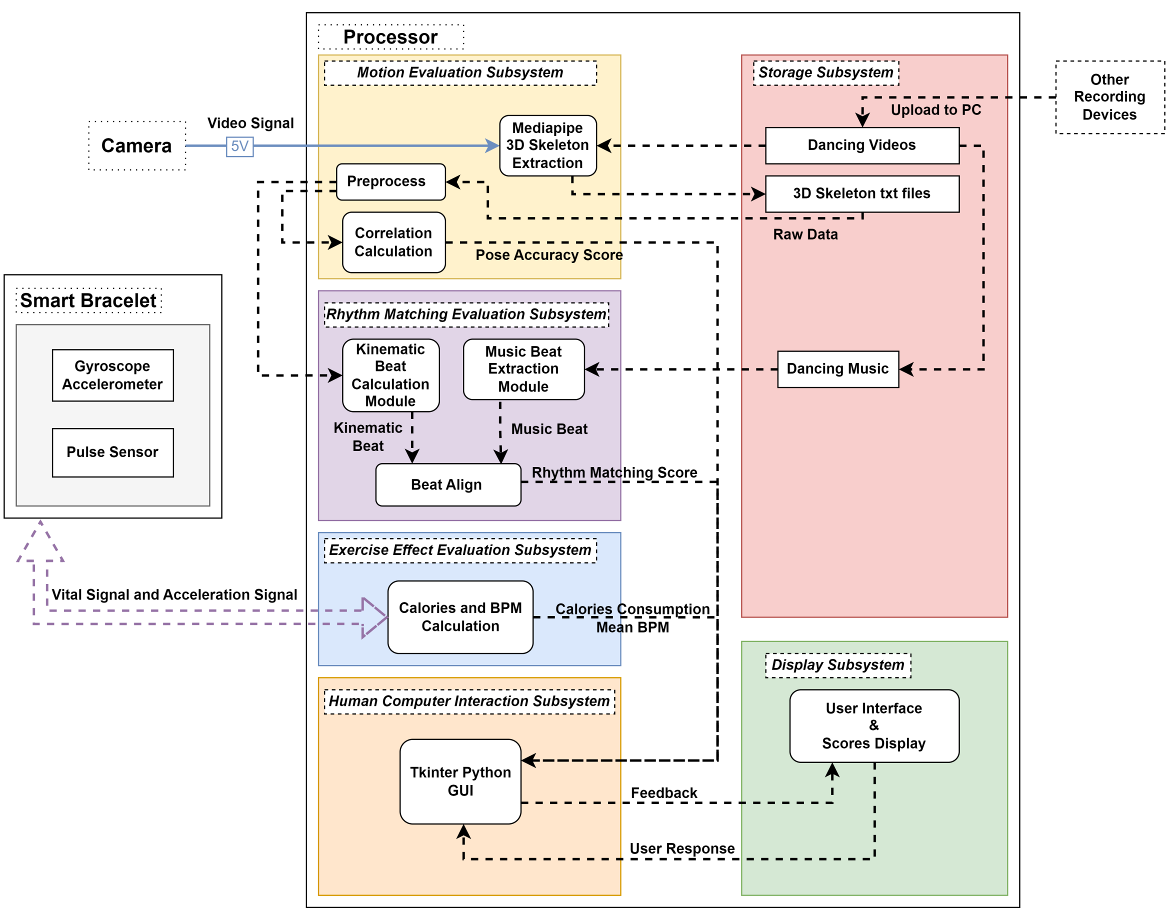

Figure 1: Block Diagram

2 Design

2.1 Bracelet

Our project dancing scoring robots gives a comprehensive and precise evaluation on users' dancing exercises. It consists of evaluation systems which are mainly software-based and exterior equipment. I am responsible for the smart bracelet implementation, which is responsible for collecting the kinematics data and health-related vital signals.

For the kinematics monitoring module, it consists of a gyroscope and an accelerometer. Signals are collected through the IMU and transmitted to the microprocessor on the board. The signals are converted from measured analog signals into digital signals through ADC. The microprocessor preprocesses the data to identify the current movement and count the number of certain movements. The processed kinematics data is transmitted to the exercise effect evaluation subsystem through a Bluetooth mesh network.

The health monitoring module consists of a Pulse sensor to collect the real-time heart rate. The heart rate collected is transmitted to the Exercise Effect Evaluation Subsystem through the Bluetooth network. This vital signal monitors the user's health and gives the necessary information for a personalized exercise effect evaluation process.

2.1.1 Embeded Design

The bracelet contributes to the dancing scoring system by collecting the vital signal and kinematics signals in an instant-time manner. In order to meet the functional requirements, the bracelet mainly consists of 1 microprocessor, 2 sensors, and 1 communication module. The Accelerometer contributes the kinematics signals and the Pulse sensor contributes the vital signal. The wire connection diagram is shown in Figure 5 below. In the Figure below, only the core chips are presented with some pins unconnected. All those pins are well-soldered in the modules in the real components. Above the connection line in the Figure, the port communication types are clearly stated. The power supply is 5V, where the current go through a transformer to 3.3V, then supply the power to all other electronic components. The requirements of the bracelet consist of 2 parts. First, the bracelet should meet the function of collecting the kinematics signals which are clearly enough for motion evaluation analysis. Second, the bracelet should be human-friendly, without any potential harm to the users.

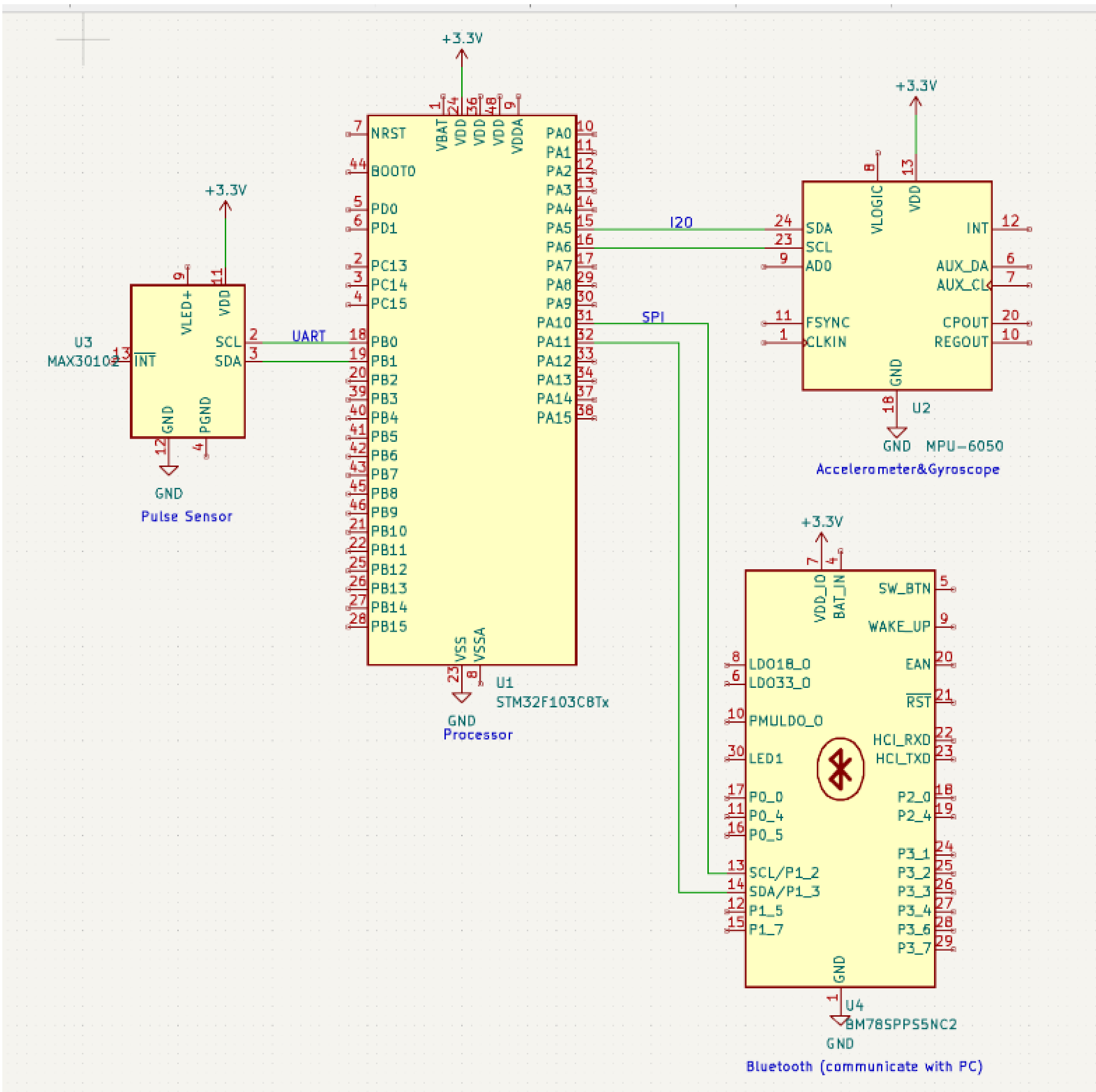

Figure 2: Bracelet Wire Connection Diagram

The accelerometer is MPU6050, which is read every 500 ms. The detailed verification and requirements are illustrated in the following section.

The pulse sensor gives out an analog signal which is created by the reflection rate change during the pulse under the human skin. This signal could be plotted out as the photoplethysmogram. The pulse sensor amped the response.

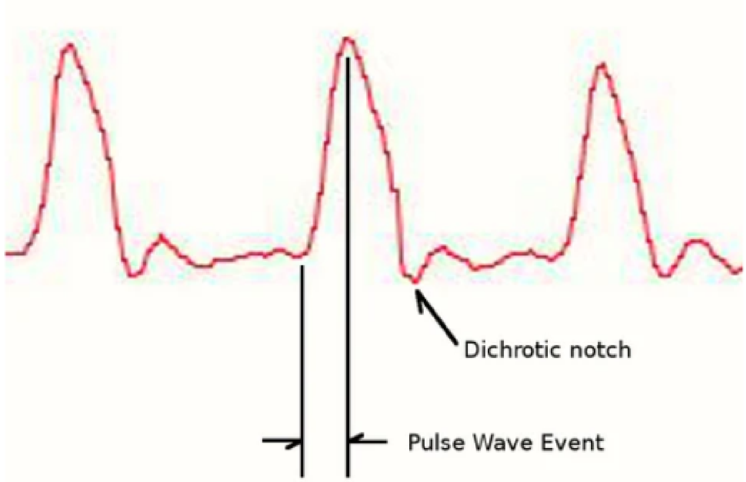

Figure 3: Sample PPG
[1]

A relative sharp upward rise in the signal value occurs as the pulse wave goes up, then go back down. The Dichroic notch of the pulse wave holds a larger amp than the upward wave. Thus, a significant rise directly after the dichroic notch would happen. In this case, our code is designed to avoid double-counting the pulse.

2.1.2 Physical Arrangement

In order to get the physical dimension of the whole bracelet, firstly we list out the electronic modules' dimensions as shown in Table 2.1.2 below. Then we decide the relative layout of the total modules. Based on the sensor working environment, the pulse sensor must be on the side facing the skin. From the aspect of human-machine interaction, in order to keep the heat transfer rate as large as possible, the battery is placed on the top. The total layout is shown in Figure 4. Thus, we could get the minimum dimensions inside the bracelet shell: 62*35.7*17, with the unit of millimeter.

Modules	Length (mm)	Width (mm)	Height (mm)
STM32	22.86	53.34	2.00
CC2541	35.70	15.20	2.00
MAX30100	Diameter = 16.00		2.00
TPS63020	24.00	34.00	4.50
Battery	62.00	62.00	8.50

Table 1: Bracelet Component Dimension

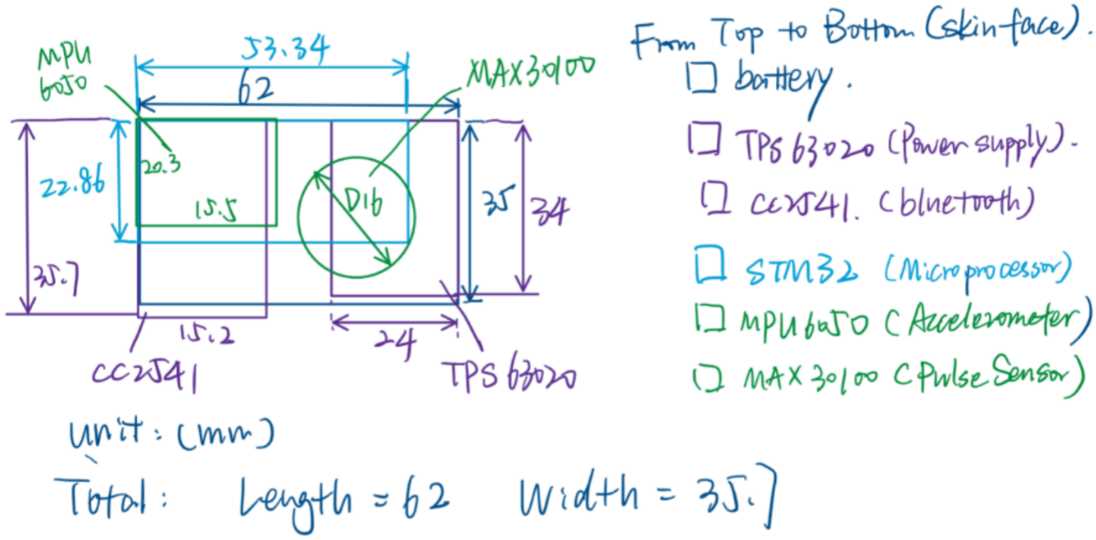

Figure 4: 2D Assembly Layout version 1

The first version of the bracelet is 3D printing the bracelet shell with PLA, the dimensions of the outer shell would be around 65*60*70, as Figure 5 shows. This version of the bracelet takes the large space that Dupont wires would take, with low space efficiency and a relatively rough human-machine interaction. The most significant progress made in this first version is that it helps to confirm the way of the shell-belt connection mechanism.

In order to improve the space efficiency and the human-machine interaction, I tried to cut off all the Dupont wires(Figure 6). The soldering process is time-consuming and causes severe problems with the components' functionality. One microcontroller is burned out, and the Bluetooth chips are seriously damaged due to the high temperature of the welding pen.

Figure 5: 3D printed bracelet shell, the first version of the bracelet.

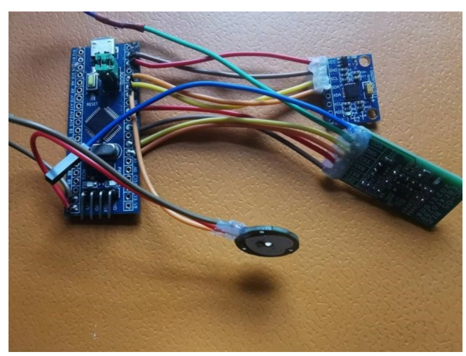

Figure 6: Soldered components, the second version of the bracelet.

The third version of the bracelet (Figure 7 and 8)shows a compromise between wearability and functional stability. The electronic components connect through Dupont wires rather than welding. The shell of the bracelet is built by PMMA, which is fabricated through a laser-cutting process. The connection mechanism and dimensions between the shell and the belt carry on the first version. The final dimension of the bracelet is around 45*50*60 (mm).

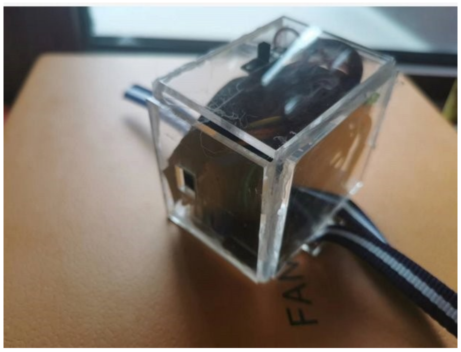

Figure 7: The third version(front view)

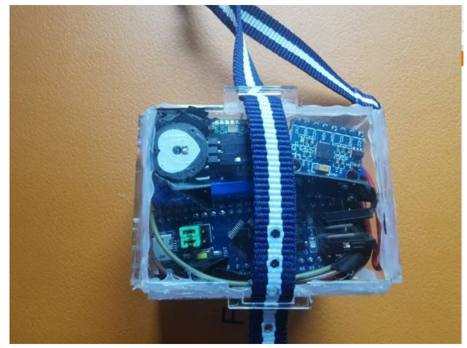

Figure 8: The third version(bottom view)

2.2 Motion Evaluation Subsystem

For the motion evaluation subsystem, our main goal is to calculate the pose accuracy of the user compared to a standard dance video, which could either be uploaded or retrieved from our database. Our steps for the motion evaluation subsystem are based on the signal processing of quaternionic data [2].

2.2.1 Global Synchronization

The original video will start recording before the music starts, the time between the start of the recording and the start of the music is redundant and needs to be cropped so that the video can correspond to the original music soundtrack. Moreover, the site environment is not quiet and should contain many disturbances, thus, the recorded music piece might not perform well in beat extraction with many background noises. That is the reason why we need to do the synchronization and replace the raw audio track with the original one. This synchronization will also be useful to align the acceleration data from the smart bracelet.

There are many algorithms for alignment, like the basic Dynamic Time Warping (DTW) algorithm, which make cross-media alignment possible [3], for example, music to note, music to lyrics, music to video motion... Here, since we are aligning two recordings of the same music, we can use Cross-Correlation (GCC) to estimate the time delay. This method was mainly used for solving the differences-of-arrival (TDOAs) problems between signals received at an array of sensors [4], of which the essence is also to handle the same music sequence with different offsets.

Basically, the procedure is divided into three steps:

Firstly, a converter will extracts recorded audio (with background noise) from the video and stores it in the current folder.

Secondly, the generalized cross-correlation is used to find the delay. The principle of GCC is simple: Assume we have two audio signals T and B (which might have some insignificant differences) with a different offset, keep the signal B unchanged, slide the signal T from left to right, and calculate the same change trend of B and T at each sliding step. If the same change trend of B and T reaches the maximum when sliding to the nth step, it means that the time difference between B and T is n steps (i.e., n sampling points), then shift T by n sampling points to align it with B.

In statistics, we use the covariance $Cov(X, Y)$to describe the degree of similarity and difference in trends between the two variables, and n indicates the correlation. In signal processing, for two discrete signals f_i and g_i, the cross-correlation function can be defined as:

$$R_{fg}(m) = \sum_{t=-\infty}^{\infty} f^*[t]g[t+m] \tag{1}$$

For continuous signals f(x) and g(x), the cross-correlation function can be defined as:

$$R_{fg}(m) = \int_{t=-\infty}^{\infty} f^*(t)g(t+m)dt \tag{2}$$

So now, for audio signals B and T, we calculate the distribution of their correlations at different m, and the m at the biggest correlation is the time difference between these two signals.

Since both Python and MATLAB provide functions of cross-correlation, we can easily realize this algorithm. Here I use *scipy.signal* in Python as an example:

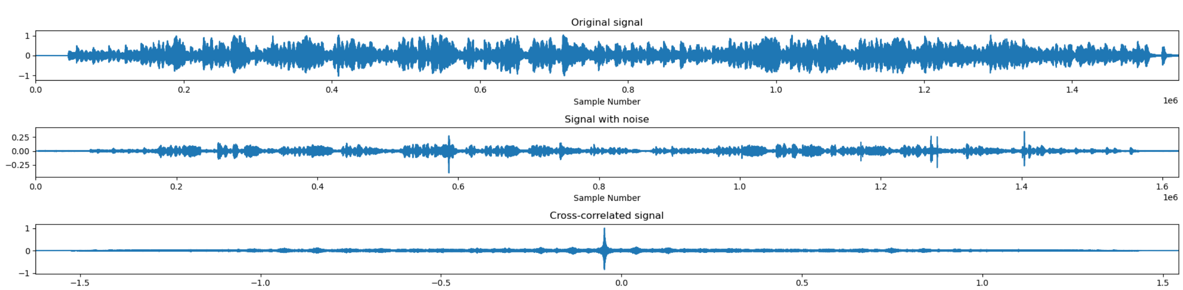

Figure 9: Music Alignment using GCC

The first plot in Figure 9 is the original audio waveform, the second plot is recorded audio with noise in the background, and the third one shows the correlation of different shift times, the highest correlation, which means two signal has the highest overlap, relates to the lag of recorded noisy signal(the index is sample number). The lag time can be calculated by dividing the sample rate of the audio signal.

In this sample, the time delay is about 2.1275736961451246 seconds.

Last, adjusts the recorded video to the correct offset calculated before, it's unnecessary to plug back the original audio because we only need the motion data from the video at the correct timeline. However, we can plug it back to verify the correctness of the offset.

2.2.2 3D Pose Extraction

In order to reduce the cost of our overall system, we decided to use Mediapipe to extract the 3D skeleton of the video. In our program, we record the user's 33 joints' 3d coordinates for every frame and save those data into a text file. Then, transform this text file into a *numpy.array* in the shape of (n,33,3), where n represents the number of total frames.

2.2.3 Pre-Filtering

Since our 3d pose extraction method might have some errors, especially at the time when quick movement happens. In order to reduce the high-frequency tracking noise, we apply a Gaussian window in the frequency domain [2]:

$$G(u) = e^{-2(\frac{u}{\sigma \cdot T})^2} \tag{3}$$

Here we choose σ as 0.1 of the video's fps, which is a proper coefficient to smooth the pose movement curve properly. The figure below shows how this filter smoothes the original noisy curve. The blue curve represents the original joint's movement and the orange curve is the curve after filtering.

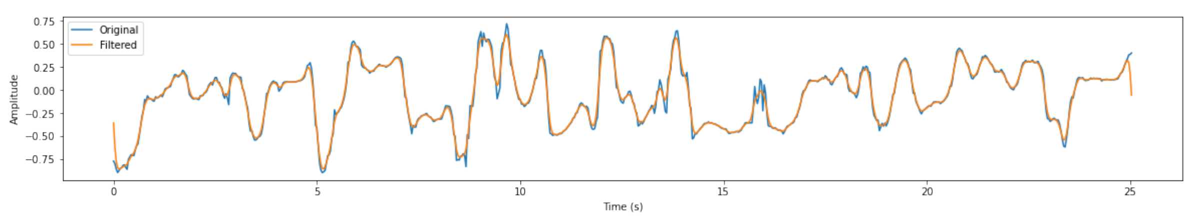

Figure 10: Gussian Filter

2.2.4 Fast Spatial Alignment

The first step for comparing the users' 3D dancing information with the standard using pre-processed 3D skeleton information is to do a rigid transformation. The goal of this step is to minimize the effects of the variance of the heights and positions between the user and the dancer in the standard video. In this way, the source of the difference can be limited to the motions of evaluating subjects.

As can be viewed from the vivid illustration in Figure 11, we will scale the user and move his/her centroid to align as closely as possible with the standard by

$$\mathbf{p}' = k\mathbf{p} + \mathbf{d},$$

, where $\mathbf{p}$ is the original 3D skeleton information, and $\mathbf{p}'$ is the aligned information. The scale is calculated as the square root of the proportion between the absolute values of the standard video dancer's 3D skeleton data $\mathbf{P}$ and the user's 3D skeleton data $\mathbf{Q}$ as

$$k = \sqrt{\frac{|\mathbf{P}(j,t)|}{|\mathbf{Q}(j,t)|}}.$$

The shift of position $\mathbf{d}$ is calculated by the difference between their centroids

$$\mathbf{d} = \mathbf{c}_p - \mathbf{c}_q,$$

where

$$\mathbf{c}_p = \frac{1}{J \times T_p} \sum_{j=1}^{J} \sum_{t=0}^{T_p-1} \mathbf{p}_j(t), \mathbf{c}_q = \frac{1}{J \times T_q} \sum_{j=1}^{J} \sum_{t=0}^{T_q-1} \mathbf{q}_j(t),$$

with J denoted as the number of joints (in our case 33), T_p and T_q are the numbers of the frame for $\mathbf{p}$ and $\mathbf{q}$ respectively.

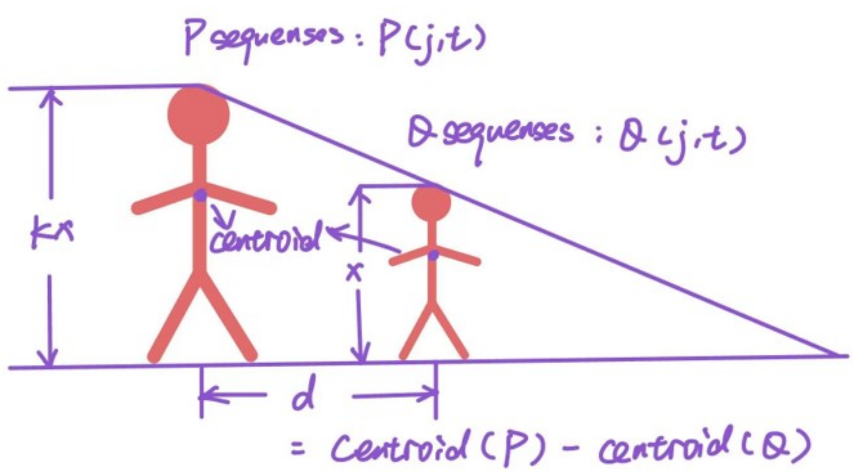

Figure 11: Illustration for fast spatial alignment.

2.2.5 Local Temporal Synchronization

After we align the static of the user and the standard video as best as we could, we start to do local temporal synchronization. This is to avoid situations in which users get extremely low scores when they are only several beats behind since for the motion evaluation subsystem, we are only comparing how well their poses are. To do this, we deploy a dynamic programming algorithm dynamic time warping (DTW).

2.2.6 Covariance and Correlation Calculation

We are ready to do our covariance based on the idea of convolution

$$\mathcal{C}_{\text{total}}(\tau) = \frac{1}{J \times T} \sum_{j=1}^{J} \sum_{t=0}^{T-1} \mathbf{P}(j,t)\overline{\mathbf{Q}(j,t)}.$$

The $\mathbf{P}$ and $\mathbf{Q}$ vectors are normalized by deducting the average to bring us back to the correct origin.
We take the maximum of $\mathcal{C}_{\text{total}}(\tau)$ as our final covariance

$$\mathcal{C}_{\text{total}} = \max \mathcal{C}_{\text{total}}(\tau), \tau \in [0, T-1],$$

and the final correlation is further obtained by

$$\text{Correlation}(\mathbf{P}, \mathbf{Q}) = \frac{\text{Covariance}(\mathbf{P}, \mathbf{Q})}{\sqrt{(\text{Covariance}(\mathbf{P}, \mathbf{P}) \times \text{Covariance}(\mathbf{Q}, \mathbf{Q}))}}.$$

The pose accuracy, i.e., the motion evaluation score, is the calculated correlation.

2.3 Rhythm Matching Evaluation Subsystem

Music and dance are closely intertwined and have a symbiotic relationship. As you can feel, many dance performances have a strong correlation with the beats, like people are born to dance involuntarily to the beats. And with certain movements being performed on specific beats or accents, the audience can receive the emotions and messages that are conveyed by the dance. So, based on this high correlation between music and dance, in recent years, many researchers are using AI to automatically generate dance motions, from which we can also see the importance of rhythm in dance.

As a part of the dancing scoring system, the rhythm-matching evaluation subsystem is quite essential. In general dance competitions, evaluating whether the dancer can keep up with the beat and rhythm of the music is also one of the final criteria for the judges' scores. In other words, the beat can be regarded as the pulse of the music, since the beat helps to determine the timing, synchronization, and musicality of a dancer's movements.

In this subsystem, we will check how well the dancer is dancing on the beat.

2.3.1 Relative Joint Positions

Even though the absolute joint positions can be used to do the pose matching, however, every joint's absolute positions include the global motion, which does not actually represent the motion of joints.

$$P_j(t) = P_j^{ab}(t) - P_{torso}^{ab}(t), j = 0, 1, 2, 3, ..., 32 \tag{4}$$

2.3.2 Relative Joint Velocities and Kinematic beat

Since our video's fps is 30, the time difference is quite small, so Δt can be used to estimate δt. We can get the velocity by:

$$v_j(t) = \frac{\delta P_j(t)}{\delta t} \leftarrow P_j(t) - P_j(t-1), j = 0, 1, 2, 3, ..., 32 \tag{5}$$

Here we will use the calculated velocity to find out the local minima as our kinematic beat for evaluation.

2.3.3 Music Beat extraction

After doing a lot of literature reading, beat extraction methods are varied, however, many are based on the detected onset [5] and frequency. Thus, I decided to use a simple way to extract beat, that is, using the audio signal processing library called *librosa*. The onset and beat submodule can easily help us to visualize the detected onset and beat on the timeline of the music. The figure 12 shown below is the Mel spectrogram and the Onset strength & Beat of the sample music piece:

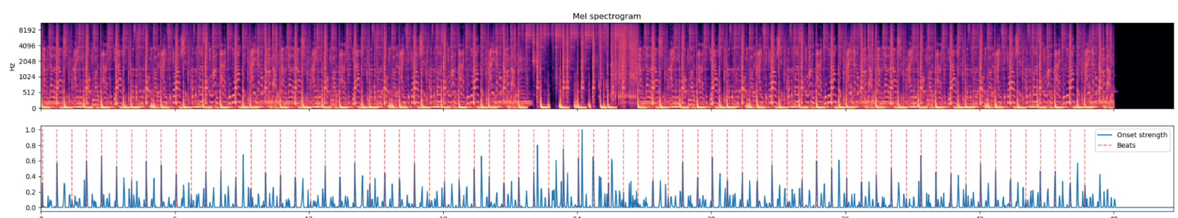

Figure 12: Mel Spectrogram and the Onset Strength & Beat

2.3.4 Beat Align Score

Many researchers have started to use neural networks to generate dance motion. Li [6] designed an AI Choreographer based on music to generate 3D dance, in the paper, in order to do the self-assessment, they propose a Motion-Music Correlation method to see how well the motions they generated are, compared to the random motion. This method inspired me how to actually do the rhythm-matching evaluation. To do the beat alignment between music and dance, they introduced a concept called "kinetic beat", which is the local minima of the kinetic velocity curve [6]. And the Beat Alignment Score [6] is defined as:

$$BeatAlign = \frac{1}{m}\sum_{i=1}^{m} exp(-\frac{min_{\forall t_j^y \in B^y}||t_i^x - t_j^y||^2}{2\sigma}) \tag{6}$$

Here, $B^x = \{t_i^x\}$ is the kinematic beats, and $B^y = \{t_j^y\}$ is the music beats. σ is used to normalize different FPS [6].

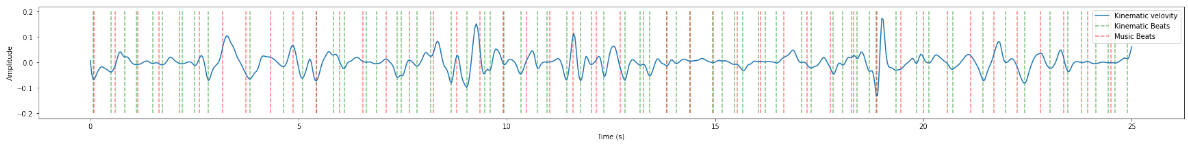

Figure 13: Beats Alignment between Music and Generated Dance

Figure 13 visualizes the process of the beat align algorithm. The blue curve is the x-axis of joint 19 of one video, and the green lines represent the kinematic beats are local minima of the kinematic velocity. The red lines are the music beats.

2.4 Exercise Effect Subsystem

Energy consumption is also part of our dancing evaluation progress, with the aim of bringing out the positive effects of dancing on the body's strength and coordination. Heart rate and acceleration data play significant roles in reflecting energy expenditure. The heart rate reflects the cardiovascular response to exercise, with higher heart rates indicating increased exertion. These data, captured by sensors in our bracelet, provide information about movement patterns, intensity, and changes in body motion during physical activities. And this subsystem will go from these two kinds of data, and generate the dancer's energy expenditure estimation.

2.4.1 Heart Rate

After comparing a lot of methods that calculate energy consumption from heart rate data during the exercise, the equation that used the percentage of Heart Rate Reserve was adopted.

$$HRR = 100[(activityHR - restingHR)/(maximalHR - restingHR)]$$

This method does not require individual calibration for each participant and can provide rapid predictions[7].

Women

Low activity level: 0.744 + 0.0216 (HRR) + 0.00699 (weight) + 0.00102 (HRR) (weight)
High activity level: 0.165 + 0.0688 (HRR) + 0.02666 (weight) + 0.00050 (HRR) (weight)

Men

Low activity level: 0.449 + 0.0627 (HRR) + 0.00743 (weight) + 0.00100 (HRR) (weight)
High activity level: 1.044 + 0.0250 (HRR) + 0.01088 (weight) + 0.00177 (HRR) (weight)

Figure 14: Caption

2.4.2 Acceleration Data

For energy expenditure prediction, I employed a CNN with six convolutional layers. This architecture was chosen to capture relevant features and patterns from the acceleration data. The CNN model was followed by a dense layer for learning complex relationships and a linear activation layer for regression. Using the regression, enabled the model to make accurate predictions of energy expenditure based on the input acceleration data. Since the data set from the Internet used for model training is 50 Hz, while our bracelet operates at a lower frequency of 2 Hz, I need to align the data. I performed linear interpolation for upsampling on the bracelet acceleration data. Due to the presence of max-pooling layers in the model, I believe the discrepancy in the predictions would be minimal.

2.4.3 Combination

Not every single prediction can respond to the whole-body reflection in the dancing. Therefore, the combination of these two kinds of data can help us establish a more convincing result. I found a combination model based on a branched equation[8]. This model aims to dynamically allocate weights to the two predictions in order to achieve an accurate and robust estimation of energy expenditure during dance performances.

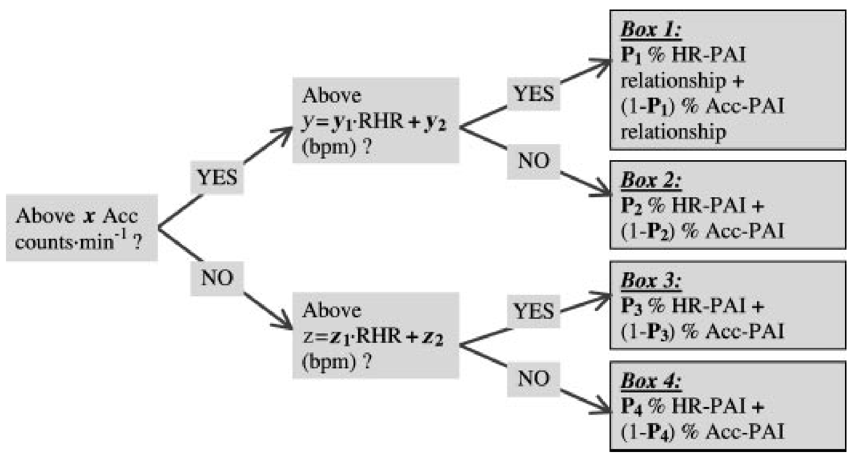

Figure 15: Caption

The advantage of using a branched equation for weight allocation is its flexibility and adaptability. It enables the combination model to be generalized, retaining some accuracy even if dynamic individual calibration information was lacking.

2.5 Human Computer Interaction Subsystem

In order to integrate all our subsystems, we use Tkinter to design a GUI. Users can access different parts of our subsystem through this GUI. The interface of GUI is shown in Figure 16.

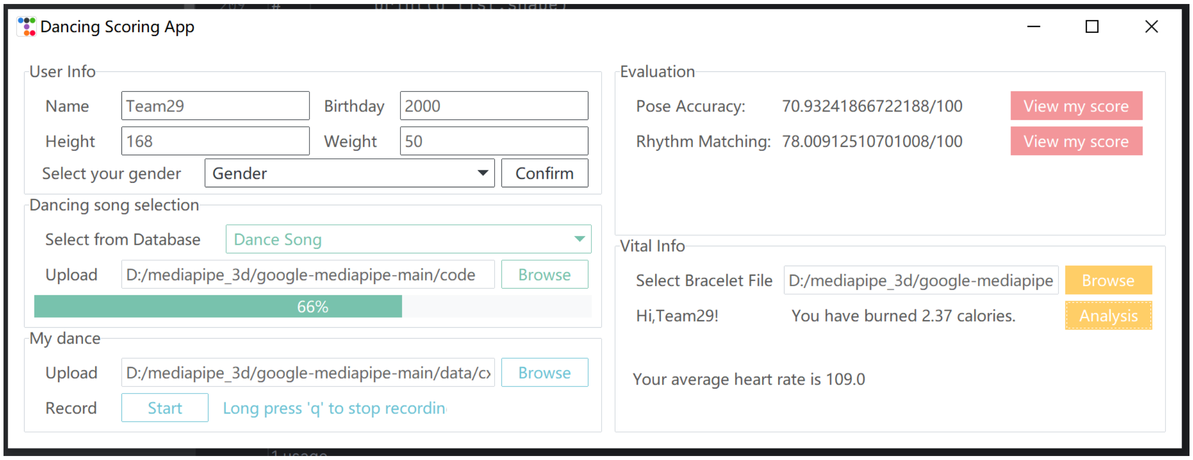

Figure 16: GUI Interface

The first part of GUI is the User Information, including name, birthday (age), height, weight, and gender. This information will be taken into consideration when evaluating the exercise effect later. By clicking the confirmation, your information will be saved in our program, and you can change this part whenever you want. In the second frame of GUI, you can either choose the standard song from our existing database or upload the video you want to estimate by yourself. If the user chooses the song from the database, the preprocessing time for the standard video will be saved.

The frame on the left bottom is the place to upload your own recording, the user can either select from the computer or record through this interface. This browsing function makes it possible for people who don't have a computer right by their side to record but are able to use the other device to record the video. They can do the recording at any time and in any place if they have their phone or camera with them, and upload to see their scores when they can use the computer. Also, this flexibility also shows in the hardware part. We do not require the user to buy the specific camera, which saves a lot of money as well as saves time for setting up.

The frames on the right side show our evaluation of different subsystems. The upper frame contains information on pose accuracy and rhythm-matching accuracy. And the frame at the bottom contains vital information about your dance performance. You need to upload the output text file produced by the bracelet, then click the "Analysis" button to see your average heart rate and the calories you burned during the dancing.

3 Verification

3.1 Verification for the bracelet

The requirements and the verification approaches for the bracelet are listed below: 1. The kinematic sensors should record the acceleration signals that could reflect certain body movements with peaks. 2. The system could give the average heart rate during the exercise. 3. The communication between the bracelet and the computer should work steadily at a distance of 3 meters. The working temperature of the bracelet should not be above 40 Celsius.

As Figure 17 shows, here is a calculated acceleration wave plot from real-time testing. The accelerometer shows the acceleration in the three dimensions, while our verification only cares about the total acceleration. Even without cutting off the noise, our signal shows a significant correlation between the sudden motion and the acceleration peak.

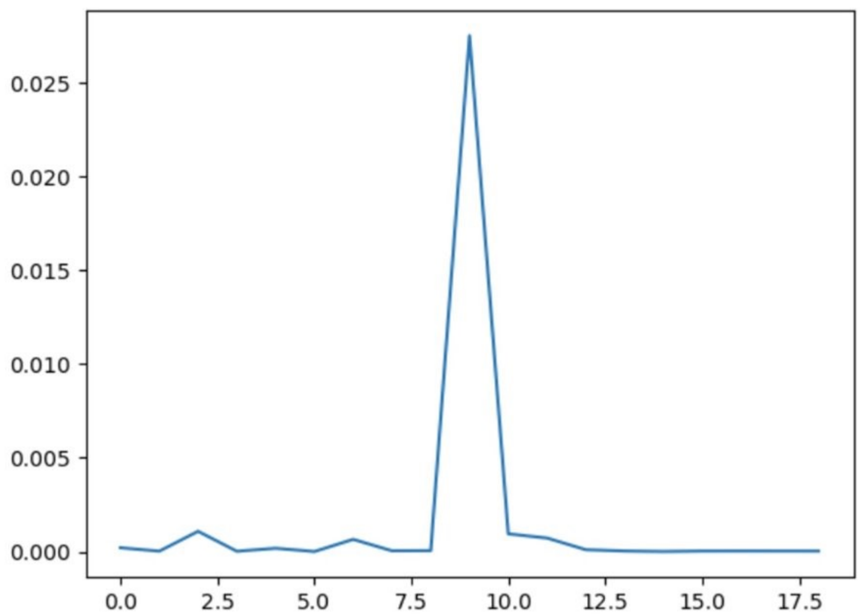

Figure 17: Sample Acceleration Calculated Result

From the bracelet output log, we could see the BPM is a bit above the normal range: 60-120. However, it is still reasonable due to the exercising effect increasing the heartbeat rate. At the same time, we use an Apple Watch on the other wrist, the bpm is around 130, which shows the validity of our bpm measurement. Similarly, we could see the temperature collected through MPU6050 is around 30 Celsius, which is a very safe temperature both for the electronic components and the user.

```
[2023-05-14 20:10:47.865]# RECV ASCII>

 acceleration:  -14496  -5522   -470  gyroscope:     -90   -136     1
temperature:    27.63
 The current AD value = 0x0820

 The current AD value = 1.671753 V

 Signal = 516

 BPM=144

 IBI = 320

[2023-05-14 20:10:48.309]# RECV ASCII>


[2023-05-14 20:10:48.371]# RECV ASCII>
 acceleration:  -15074  -5558   1272  gyroscope:     -34    -16     10
temperature:    27.66
 The current AD value = 0x07FF

 The current AD value = 1.642749 V

 Signal = 511

 BPM=144
```

Figure 18: Sample Bracelet Ouput Log

In the drive of our Bluetooth part, the microcontroller checks the connection between the Bluetooth and the PC every 5 seconds. During each run of our test, the microcontroller never sent a 'connection failure' message, thus the validity of communication is stable.

3.2 Verification for the motion evaluation subsystem

The effectiveness of the motion evaluation subsystem has been fully verified through a large amount of empirical studies.

3.2.1 Verification for the 3D skeleton information extraction

When running the motion evaluation function, the 3D skeleton information of both the standard video and the dancer is displayed in real-time simultaneously with the video. As can be viewed in Figure 19, the skeleton indexes well align with the dancer in the video.

Figure 19: Extraction of 3D skeleton information.

3.2.2 Verification for the accuracy of Global Synchronization

The index of the highest correlation we calculate in part "refine video" is the index of the sample we need to shift since the x-axis is the samples of the music piece. We can compare two waveforms after synchronization to verify. However, since there are some noises in the background which might result in very different waveform plots, we can also verify it by just hearing.

We shift the noise audio to the right offset and slice it into the same length as the original audio. We use *IPython.display.Audio* to display these two audios, and play both audio clips at the same time. Then they will be identified by people to see if there's any time delay.

When two audio clips are playing, I will mute one channel with a louder sound to see how well the other channel continues. If the music sounds like it skips a few notes or repeats a few notes, it means the alignment fails. I repeated this procedure many times, and I cannot identify any time difference, the music went smoothly. Which means we get nice time delay information.

3.2.3 Verification for rigid transformation

We performed two sets of tests. First, as illustrated in Figure 20, we choose two videos from Produce 101, a dance show, where the two idol dancers are of similar heights and are standing at similar positions on the stage. We have calculated a scaling factor of $k = 1.0162560097646425$, which is approximately 1, and a centroid shift vector

$$\mathbf{d} = [-0.03955619, 0.00640521, 0.12179582],$$

which has very small absolute values. The result is reasonable.

As a comparison, we have the standard video with an idol dancer who is as tall as almost 2 meters, and we invited one of our classmates to do the dance and adjust the angle a bit so he appears shorter in the video. We obtain see a reasonable scale of $k = 0.7603874757671618$ based on the comparison of the two dancers performing the same pose in Figure 21. Also, we shoot the video deliberately further from the dancer (Figure 21), giving a larger absolute value in the third dimension of the centroid shift $\mathbf{d} = [-0.04265497, 0.25888725, -2.20253157]$.

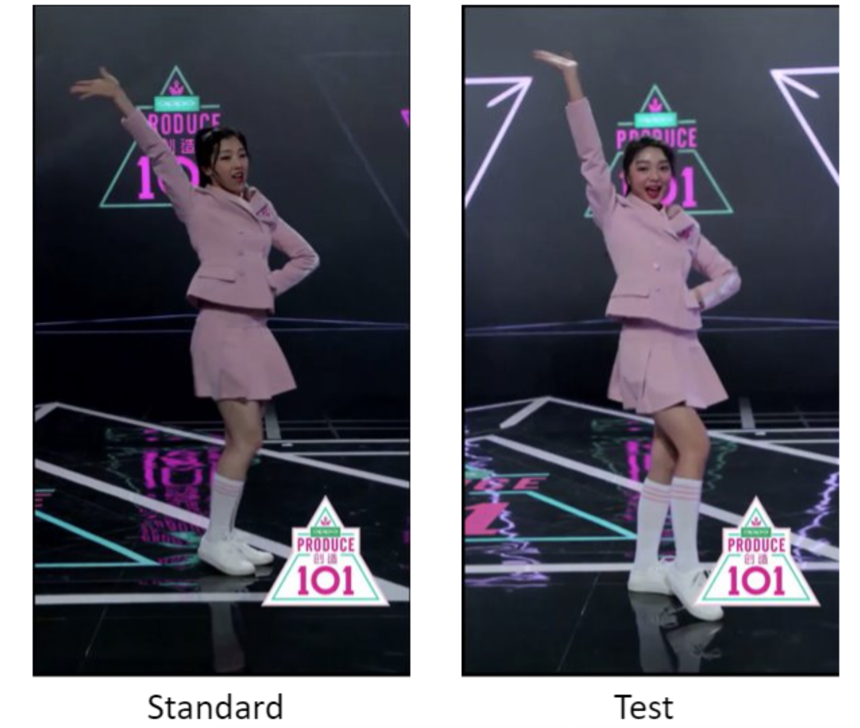

Figure 20: Rigid Transformation: Produce 101.

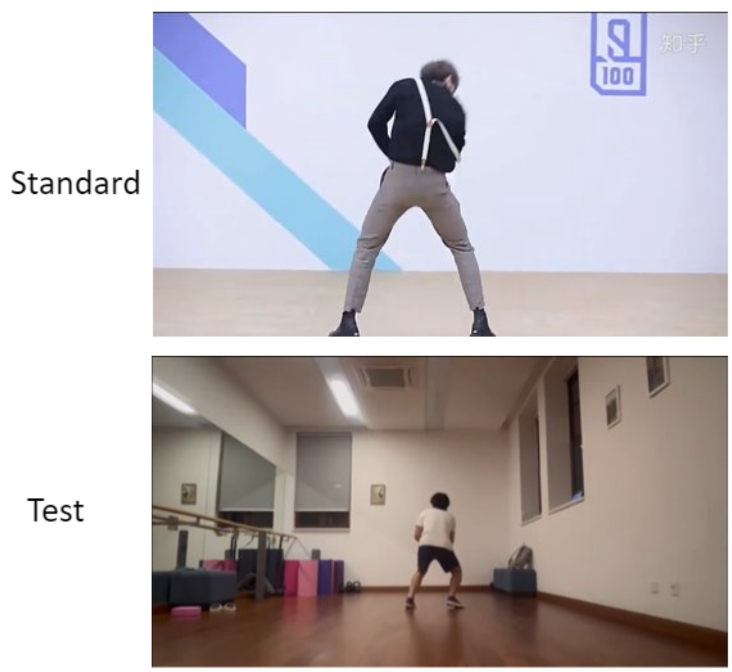

Figure 21: Rigid Transformation: Idol dancer vs our invited classmate.

3.2.4 Verification for local temporal synchronization

We have taken the dance from the idol mentioned above as standard, and a professional dancer's performance with some freestyles as test. Before DTW, the pose accuracy is only **22.12**, which is not reasonable since he is matching a majority of the poses. After DTW alignment, his score has been raised more than three times to **70.93**, which apparently makes more sense.

3.2.5 Verification for the pose accuracy scores

For this part, we take the idol dance as standard and included three different tests with great variations Figure 22. Test 1 is performed by one of our classmates who is a beginner dancer. He took ten minutes in total to view, practice, and perform and has been given an expected score of 50. Test 2 is performed by a professional dancer with a lot of practice, but also with a lot of his own styles in the dance. Thus, he is given an expected score of 75. Test 3 is made according to the skeleton extracted from the standard video and virtually covered. Since there might be some difference in the 3D information extraction algorithm, the third test is given an expected score of 90.

Figure 22: Experimental setup for the verification of the pose accuracy scores.

As can be viewed from our experimental results in Table 3.2.5, the motion evaluation subsystem is able to generate pose accuracy scores of more than 90%.

Test Number	Expected Score	Actual Score
Test1	50	45.22
Test2	75	70.93
Test3	90	91.39

Table 2: Results for the verification of pose accuracy scores.

3.2.6 Verification for efficiency of the motion evaluation subsystem

A 15 seconds long dancing video could be processed and evaluated within 1.5 minutes.

3.3 Verification for the rhythm matching evaluation subsystem

3.3.1 Verification for the accuracy of the detected beat

Since the beats' actual time is hard to manually record, and based on the fact that the interval between each beat is almost totally identical, the accuracy can be verified by using the cumulative counting method. To be specific, for one music piece, we can count the number of beats for the music piece (if the music is too long, we can only count the first thirty seconds), and compared it with the detected beats number in the length to get the difference. Because the manual count of beats may have errors, at least three manual counts are required. If all three times' results are the same, we can assume this result is ground true, else, we will have more tries to get a correct result.

Here is our test table, we randomly choose three pieces of music for testing, since none of these audios are very long, we will count the number of beats in the entire audio:

Music	mMH1.wav (48s)	mLH4.wav (34s)	mLH5.wav (31s)
Manually count 1	71	64	64
Manually count 2	72	64	64
Manually count 3	71	64	64
Detected count	71	64	63
Accuracy	99.5792%	100%	98.4375%

Table 3: Beat Accuracy Test

As Table 3.3.1 shows, the results of our beat calculation basically match our manual count, leaving aside the fact that there may be some differences that whether the initial and final beat should be included, there is almost no error in our beat extraction.

3.3.2 Verification for whether the Beat Align Score is reasonable

Here I use the data from the famous talent show: Produce 101. Girls in this show have performed the same theme song and are recorded individually, which is a good resource to test our algorithm. I randomly choose some girls' performances, and they were given the rates and ranks according to their performances by the professional judges. I test their performance with the beat align algorithm, and the results are shown below. Their ranks and rates highly correlate with the beat align scores, and it is reasonable that the dancers with higher ranks show a better sense of the rhythm.

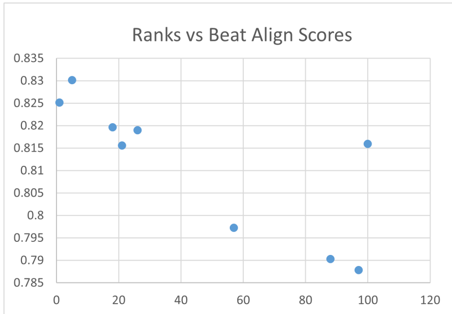

Figure 23: Ranks vs Beat Align Score

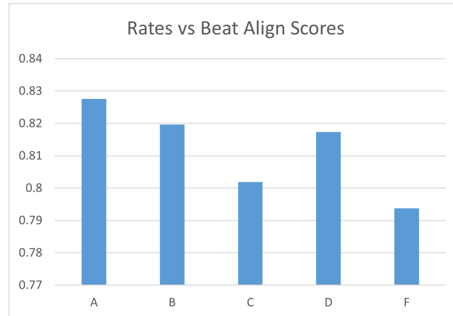

Figure 24: Rates vs Beat Align Score

3.4 Verification for the exercise effect evaluation subsystem

Due to the limitations that we cannot directly measure the exact energy expenditure during dance performances, our design verification focused on qualitative analysis to assess the reasonableness and consistency of the output results. The following steps were taken to validate the effectiveness of our approach:

We collected heart rate and three-axis accelerometer data from a group of dancers during their dance routines. Although we did not have direct measurements of energy expenditure, we used the collected data as a proxy for the intensity and effort exerted by the dancers.

The output results from the heart rate variation analysis and three-axis accelerometer data analysis were qualitatively analyzed for reasonableness. I examined the trends, patterns, and variations in the predicted energy expenditure throughout different times input.

I also compared the output results with some existing studies and established guidelines for energy expenditure estimation during physical activities. And the results were also highly similar. This comparative analysis helped us make our design more accurate and convincing.

3.5 Verification for Human-Computer Interaction Subsystem

3.5.1 Verification for whether the GUI can show evaluation from different aspects directly

Our GUI integrates all the subsystems and shows them together. The GUI interface is clear and neat and easy for users to find their evaluation in the right frame of the GUI, including the pose accuracy score, rhythm-matching score, and exercise effect information.

3.5.2 Verification for whether the GUI is user-friendly

We ask our team members and classmates to use our GUI without giving any tutorials, and almost all of them know how to use our system through this interface. Some scores require the user to upload or fill in something to execute. If the user forgets to do some steps before evaluation, it will raise the error message to tell the user what to do as shown in Figure 25

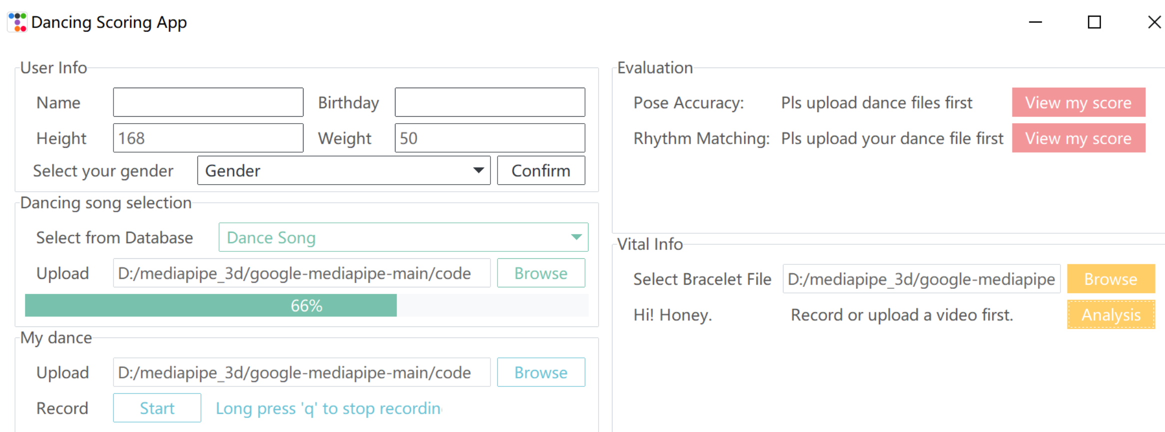

Figure 25: Hint messages in GUI

4 Cost

4.1 Cost Analysis

Costs from hardwares and external devices:

Subsystem	Item	Source	Cost(CNY)
Bracelet (Electronic)	MPU6050	Taobao	7.70
	STM32f103c8t6	Taobao	77.00
	MAX30100	Taobao	6.50
	TPS64020	Taobao	18.70
	5V battery	Taobao	26.50
	CH340	Taobao	4.50
	CC2541	Taobao	18.01
	Wires	Taobao	5.62
	Soldering	Electronic Lab	0.00
Bracelet (Mechanical)	Strap	Taobao	24.00
	3D printing	Design Lab	0.00

Deep Camera	Intel RealSense	Taobao	920.00
Motion Camera		ZJUI Lab	0.00
Total			1108.53

4.2 Labor

Our fixed development costs are estimated to be 12.5CNY/hour, 10 hours/week for four people. We consider approximately 60% of our final designs this semester (8 weeks), neglecting the central server, mesh network optimization, and partnerships with NGOs:

Labor: (For each partner in the project)

Assume a reasonable salary,

$$(12.5\text{CNY/hour}) \times 10\text{hours/week} \times 8\text{weeks} = 1000\text{CNY}.$$

5 Conclusion

5.1 Accomplishments

By developing the dancing scoring robot, we have successfully built a robotic system that is easily accessible and affordable since users can use their own PC to run our system rather than having to do clumsy installations like existing products, and our bracelet is very cheap to build compared to those on the market. Our dancing scoring robot is able to evaluate dancers' performance based on multiple evaluation aspects and using flexible evaluation standards, including those given by users themselves. Last but not least, our dancing generates reasonable and timely scores, which greatly facilitates the process of evaluating dances for entertainment in everyday life.

The code can be found at: https://github.com/xiaohan1129/DancingScoring

5.2 Uncertainties

Though our system has been proven to be robust across a large number of tests, there still exist some uncertainties. First of all, our bracelet has rather low wearability, making the measured heart rates and accelerations might not be always accurate. Besides, the 3D skeleton information extractions can fail to effectively extract the correct joint positions if the dancer wears clothes of the same color as their backgrounds.

5.3 Ethics considerations

Our dancing scoring robot raises several ethical concerns related to fairness and user privacy. In this response, we will examine these concerns through the lens of the Institute of Electrical and Electronics Engineers (IEEE) and Association for Computing Machinery (ACM) Code of Ethics [9], [10].

According to the IEEE Code of Ethics [9], engineers are required to "treat all persons with dignity and respect and avoid illegal discrimination or harassment." The dancing scoring robot must ensure that all participants are treated fairly, regardless of factors such as age, gender, ethnicity, or physical ability. The algorithm used to score the dancers should thus be designed to eliminate biases and be based on objective criteria such as technique, rhythm, and musicality. The scoring system should also be periodically reviewed and audited to ensure that it is functioning as intended and that any issues are identified and corrected promptly.

The ACM Code of Ethics [10] emphasizes the importance of avoiding harm and ensuring that technology is used in ways that benefit society. In the context of a dancing scoring robot, fairness would require that the robot does not cause harm to any participant or negatively impact their self-esteem. The robot should be programmed to provide constructive feedback that helps participants improve their dancing skills rather than being overly critical or punitive. Additionally, the robot should be designed with accessibility in mind, accommodating all types of dancers, regardless of their physical abilities.

The IEEE Code of Ethics [9] also states that engineers should "protect the privacy and confidentiality of their clients or employers' information, including personal information." In the case of a dancing scoring robot, this would require that any personal information collected from the participants, such as name or age, is kept confidential. Additionally, any video or audio recordings of the participants' performances should be stored securely and only used for the purpose of scoring and providing feedback.

According to The ACM Code of Ethics [10], the participants should be informed of what data will be collected, how it will be used, and who will have access to it. They should also be given the option to request that their data be deleted after usage. The robot's system, especially the cloud storage, should be designed with appropriate privacy safeguards and controls to ensure that the participant's data is not misused or exposed to unauthorized parties.

5.4 Future work

So far we run our system on our own computer, but it will be much easier for users to access if we do the whole evaluation process in the cloud, Users can simply access the website to use our services. Also, our bracelet can be further simplified into a single PCB with a button battery and be produced in silica gel to improve human-machine interaction. Moreover, if we also generate some feedback along with our scores, it will help users better understand how to improve their performance, and this can be used as an online dance tutor for people who want to learn the dance by themselves.

Future work can also be done to predict the video stream. The system can add some prediction parts for users to see whether their dancing video will be popular if they upload it on social media.

References

[1] P. S. Project, “”Pulse Sensor Amped”,” 2018. [Online]. Available: https://pulsesensor.com/pages/pulse-sensor-amped-arduino-v1dot1 (visited on 05/10/2023).

[2] D. S. Alexiadis and P. Daras, “Quaternionic signal processing techniques for automatic evaluation of dance performances from mocap data,” *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1391–1406, 2014. DOI: 10.1109/TMM.2014.2317311.

[3] W.-T. Chu and S.-Y. Tsai, “Rhythm of motion extraction and rhythm-based cross-media alignment for dance videos,” *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 129–141, 2012. DOI: 10.1109/TMM.2011.2172401.

[4] M. Cobos, F. Antonacci, L. Comanducci, and A. Sarti, “Frequency-sliding generalized cross-correlation: A sub-band time delay estimation approach,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1270–1281, 2020. DOI: 10.1109/TASLP.2020.2983589.

[5] Y. Zhu, C. M. Manders, F. Farbiz, and S. Rahardja, “Music feature analysis for synchronization with dance animation,” in *TENCON 2009 - 2009 IEEE Region 10 Conference*, 2009, pp. 1–5. DOI: 10.1109/TENCON.2009.5396035.

[6] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, “Ai choreographer: Music conditioned 3d dance generation with aist++,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 381–13 392. DOI: 10.1109/ICCV48922.2021.01315.

[7] H. Hiilloskorpi, M. Pasanen, M. Fogelholm, R. Laukkanen, and A. Mänttäri, “Use of heart rate to predict energy expenditure from low to high activity levels,” *International journal of sports medicine*, vol. 24, pp. 332–6, Aug. 2003. DOI: 10.1055/s-2003-40701.

[8] S. Brage, N. Brage, P. W. Franks, *et al.*, “Branched equation modeling of simultaneous accelerometry and heart rate monitoring improves estimate of directly measured physical activity energy expenditure,” *Journal of Applied Physiology*, vol. 96, no. 1, pp. 343–351, 2004, PMID: 12972441. DOI: 10.1152/japplphysiol.00703.2003. eprint: https://doi.org/10.1152/japplphysiol.00703.2003. [Online]. Available: https://doi.org/10.1152/japplphysiol.00703.2003.

[9] IEEE, “”IEEE Code of Ethics”,” 2016. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/10/2023).

[10] ACM, “”ACM Code of Ethics”,” 2018. [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/10/2023).

Appendix A Requirement and Verification Table

Subsystem	Requirement	Verification	Verification status (Y or N)
Motion Evaluation Subsystem	1. The system must be able to extract and compare poses from 3D views.	The 3D skeleton information of both the standard video and the dancer should be displayed in real-time simultaneously with the video and should be aligned with the videos.	Y
	2. The system must achieve a reasonable scoring of at least 90% correlation between existing dance videos and intuitions.	The difference between the pose accuracy score and expected ratings should be less than 10%.	Y
Rhythm Matching Evaluation Subsystem	1. The system must be able to synchronize the music that starts at any time.	It has been verified by playing the synchronized music and the standard music at the same time, then randomly muting one of them to identify whether the music is continuous.	Y
	2. The system must be able to extract the drums and beats of different music and achieve a minimum of 95% correctness.	It has been verified by randomly selecting 3 pieces of music and comparing the difference between the number of manually counting beats and detected beats. The accuracy can reach almost 100%	Y
	3. The system must be able to align kinetic movements with the rhythms/beats of the background music.	It has been verified by giving the beat align scores through our GUI and testing its reasonableness with dance videos of Produce 101.	Y

Bracelet + Exercise Effect Evaluation Subsystem	1. The kinematic sensors should record the acceleration signals that could reflect certain body movements with peaks.	It has been verified by conducting a 5s recording with one large-scale movement of the wrist, and the acceleration calculated output shows only 1 significant peak	Y
	2. The system could give the average heart rate during the exercise. Then give out the exercise evaluation mainly based on calories spent during the dancing.	Similar to the test above, the Bluetooth output log gives a log with the heart rate at each sampling point. As long as the calculated avg BPM is within the 60-150, it could be verified	Y
	3. The communication between the bracelet and the computer should work steadily at a distance of 3 meters. The working temperature of the bracelet should not be above 40 Celsius.	From the output log, the temperature could be easily read. At each test run, the communication is stable without failure warning.	Y
Human-Computer Interaction Subsystem	The system must be user-friendly.	1. It has been verified by asking some people to use the GUI without telling them how to use it, and most of them can use our GUI without giving any hints.	Y
	2. The system should show our evaluation from different aspects directly.	It has been verified by showing the user the right part frame of our GUI, which includes the pose accuracy score, rhythm-matching score, and exercise effect information.	Y