# A Hidden Markov Model-Based Performance Recognition System for Marching Wind Bands

Wei Jiang
Shenyang City University, Shenyang 110100, China

***Abstract*—This paper explores the automatic recognition of marching band performances using advanced music information retrieval techniques. Music, a crucial medium for emotional expression and cultural exchange, greatly benefits from the harmonic backing provided by marching wind orchestras. Identifying these performances manually is both time-consuming and labor-intensive, particularly for non-professionals. This study addresses this challenge by leveraging Hidden Markov Models (HMM) and improved Pitch Class Profile (PCP) features to automate the recognition process. The research also explores the system's performance on real-world audio recordings with background noise and microphone variations. By dividing the audio signal into frames and transforming it to the frequency domain, the PCP feature vectors are extracted and used within the HMM framework. Experimental results demonstrate that the proposed method significantly enhances recognition accuracy compared to traditional PCP features and template matching models. The study identifies challenges in distinguishing similar tonal values, such as F-major and D-minor, which affect recognition rates. Additionally, the research highlights the importance of addressing background noise and microphone variations in real-world applications. Ethical considerations regarding privacy and intellectual property rights are also discussed. This research establishes a comprehensive system for automatic marching band performance recognition, contributing to advancements in music information retrieval and analysis.**

***Keywords*—Music information retrieval; Hidden Markov Model; feature extraction; automatic music recognition; marching band performance; PCP features***

## I. INTRODUCTION

Music serves as a powerful medium of artistic expression. It enables people to express personal feelings and fulfill spiritual needs, while also fostering cultural exchange and promoting the development and integration of cultural diversity. Within the foundation of music theory, marching wind orchestra performances play a crucial role. They complement and enhance the main theme, adding depth and richness to the overall musical experience [1-3]. Despite its importance, the identification of marching band performances remains challenging and time-consuming, particularly for non-professionals. This paper aims to address this research gap by developing an automatic recognition system leveraging HMM and improved PCP features. If beautiful music lacks a harmonic backing, the overall effect will be greatly reduced [4, 5]. However, the identification of marching band performance often requires specialized knowledge and training that is difficult for non-professionals to accomplish accurately, especially in improvisation, where identification of marching band performance is even more challenging [6]. For many years, the identification and recognition of marching wind band performances are mostly done manually, which is time-consuming and laborious [7]. With the development of multimedia and network technology, the importance of music information retrieval technology is becoming more and more obvious. The traditional low-level features such as Mel frequency cestrum coefficients have limited effect in music semantic analysis, while the marching band performance, as a middle-level feature, contains rich music information, which is important for music analysis and retrieval [8]. Marching wind band performance is closely related to the emotion of music, which can help recognize and retrieve songs with similar styles. However, the system's performance on real-world audio recordings with background noise and microphone variations remains an important consideration [9].

Speech recognition technology has made significant progress in recent years, and HMM combined with genetic algorithm training has become a mainstream technology with the advantages of high recognition rate and fast response. However, the development of music recognition technology is slow and there are fewer related products on the market, mainly due to the low recognition rate [10]. The earliest music feature extraction methods used Mel-frequency cestrum coefficients, but nowadays it is common to use pitch-set files to represent music, which can more accurately represent music features [11]. With the improvement of computer performance and Internet bandwidth, as well as the development of multimedia information technology, content-based multimedia retrieval techniques have emerged [12]. In music retrieval, marching wind orchestra performance, as a mid-level feature, can effectively support music segmentation, retrieval and sentiment analysis [13-15]. Automatic marching band performance recognition techniques have attracted the attention of a large number of researchers in the field of music information retrieval. The correct recognition and sequence generation of marching wind band performances can help the segmentation of musical structures and the identification of specific melodies, and can reveal the potential emotional connections of music [16].

The Electrical Engineering Department at National Taiwan University was a pioneer in using PVP feature vectors for performance recognition [17]. Their system processes input audio signals by segmenting them into frames and converting them into the frequency domain to extract PCP feature vectors [18]. The recognition process is divided into two phases: training and testing. In the training phase, a Hidden Markov Model (HMM) is used, where each state corresponds to a specific marching wind band performance [19]. The state transition matrix represents the probability of transitioning from one performance to another, while the observation distribution

indicates the likelihood of a particular PCP feature vector being generated by a specific state. In the testing phase, the observed feature vectors and the trained HMM are used to decode the most probable sequence of marching wind band performances. The team's innovative use of the N-gram algorithm within the HMM framework significantly reduces complexity and enhances recognition efficiency. In 2003, Alexander Sheh and Daniel P.W. Ellis from Columbia University proposed a system that converts arbitrary audio signals into corresponding performance sequences [20]. The system process includes audio framing, transforming to the frequency domain by Fourier transform, then mapping out PVP feature vectors, constructing a marching band performance model, and utilizing EM algorithms to complete the recognition in the HMM framework [21]. Although the recognition rate of this system for marching wind band performance is only 22%, it is innovative in that only the performance sequence is considered without the need of temporal requirements on the performance transformation. In 2005, Bello and Pickens applied the EM algorithm under the HMM framework, introduced the music knowledge into the model, and avoided arbitrary initialization by defining the state transfer matrix, and achieved a recognition rate of 75% [22, 23]. Although Markov models have been successful in speech recognition, there are challenges in applying them to music. Music has complex acoustic variations and requires more data for training. Manually labeling the performance boundaries of long sections of music is time-consuming and error-prone. The music and acoustics research center at Stanford University proposes a method to automate the performance boundary labeling by synthesizing audio to generate training data, which significantly improves the efficiency of model parameter estimation.

The purpose of the research in this paper is to establish a complete marching band performance recognition system, using audio files as the input of the whole system, and returning the marching band performance sequences recognized by the system as the output to the user, so as to realize the automatic recognition with performance as the basic unit. The research content of this paper mainly includes the following aspects: first, feature extraction of music. Since the speech signal is time-varying rather than smooth, and the human articulatory organ muscles move slowly, the speech signal can be considered smooth locally, so the processing methods and theories of smooth processes can be introduced into the processing of speech signals, thus simplifying the analysis of speech signals. Next, the marching band performances applied during the experiments are extracted, and a Hidden Markov Model is initialized for each marching band performance, using a single Gaussian observation function during the initialization process, with the mean vector set to 0 and the covariance matrix set to 1 [24]. Next, the experimental samples are labeled. The labeling is done from a MIDI file, so the input file must be converted to the corresponding MIDI format, and finally a piece of music performance is extracted as the output of the labeling process. Next, the system is trained, using a pitch set file to represent the music file, and the labeled marching band performance is used as the base model to train the system, with as many training samples as possible, in order to give the system access to all the performance models. Finally, system testing is performed, where a correctly labeled music file is used as input to check the performance of the system. The remainder of this paper is organized as follows: Related work is given in Section II. Section III presents the theoretical background on Hidden Markov Models. Section IV describes the design of the marching band performance recognition system. Section V discusses the experimental results and analysis. Section VI concludes the paper with a summary of findings and suggestions for future research.

## II. Related Work

Initial efforts in music recognition heavily relied on low-level audio features such as Mel-Frequency Cepstral Coefficients (MFCCs). MFCCs were widely adopted due to their efficiency in capturing the timbral properties of audio signals [25]. However, their effectiveness in higher-level musical semantic analysis, particularly for complex structures like marching band performances, was limited [26]. The integration of Hidden Markov Models (HMMs) in music recognition was significantly advanced by Sheh and Ellis [27]. Their system converted arbitrary audio signals into performance sequences using HMMs, involving audio framing. Further refinement was seen in the work of Bello and Pickens [28], who applied the Expectation-Maximization (EM) algorithm within the HMM framework. Recent research has focused on enhancing the feature extraction methods to improve the recognition accuracy of musical signals. Enhanced PCP features have emerged as a crucial development, addressing tonal ambiguity and providing a more robust representation of musical content. Comparative studies have demonstrated that traditional template matching models [29], while straightforward, are often inadequate for dynamic and complex musical environments. In contrast, the combination of improved PCP features with HMMs has shown superior performance. However, a critical gap in existing research is the robustness of these systems in real-world scenarios, characterized by background noise and variations in recording conditions [30].

## III. Hidden Markov Model

A Markov model is a demographic tool extensively utilized in diverse natural language processing applications, including speech recognition, automatic lexical annotation, and probabilistic grammar analysis. If the "future" of a process depends only on the "present" but not on the "past", the process is Markovian, or the process is called Markovian.

In addition, since speech signals are time-varying rather than smooth, and since the muscles of the human articulatory organs move slowly, speech signals can be considered locally smooth. In this way, we can introduce the processing methods and theories of smooth processes into the processing of speech signals, thus greatly simplifying the analysis of speech signals.

### *A. Characteristic Representation of Music*

When analyzing audio signals, extracting and characterizing information from the time domain can be challenging due to the non-linear and often discontinuous nature of audio performance in this domain. In speech signal processing, converting audio signals from the time domain to the frequency domain is a common practice for more effective analysis. This technique is equally applicable to music signals. The two primary methods for this transformation are the short-time Fourier transform

(STFT) and the constant Q transform (CQT). Both methods convert the music signal from the time domain to the frequency domain, but they use different algorithms and computational processes to achieve this.

Fourier transform processing of speech signals, the premise is that the signal is always in a smooth state, but the audio signal of music is usually non-stationary, cannot be transformed by the Fourier transform spectrum to extract the spectral energy information, based on the assumption that the music signal is in the short-term transient conditions, that is, you can through the STFT spectral transformation, and then be able to analyze the characteristics of the signal in the frequency domain. The formula of STFT is expressed as:

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n}$$

Considering the continuity of the music signal, $x[n]$ represents the discretized representation, $w[n-m]$ represents the sliding time window, and $X_m(\omega)$ represents the transformed spectrum. Under the influence of Heisenberg's uncertainty principle, the time resolution and frequency resolution change accordingly because of the different window functions, if the function is determined and the window length is determined, both the time resolution and frequency resolution can not be changed, so it is difficult to deal with the non-smooth and mutated signals effectively, and it is suitable to deal with the slow-varying signals because of the insensitivity to the instantaneous changes. For the music audio signal used in this paper, there is only a single main theme after the relevant preprocessing, and its changes are more moderate, so the spectrum can be analyzed by short-time Fourier transform. Therefore, before extracting the PCP features from the audio signal, this paper employs the STFT to convert the time-domain signal into the frequency domain. This approach conserves computational power and aligns well with the subsequent PCP feature extraction. The flowchart of feature extraction in Fig. 1 shows the process of marching wind band performance recognition.

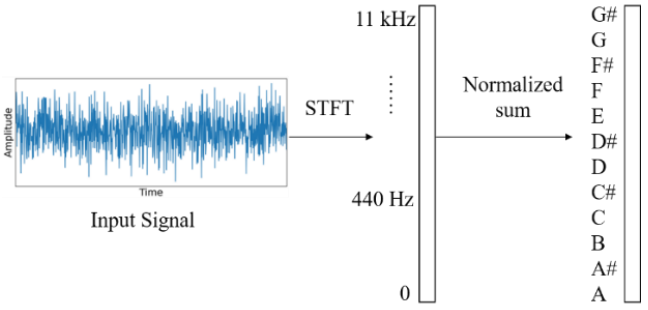


Fig. 1. Flowchart of feature extraction.

The calculation of PCP features is based on mapping frequency changes according to the twelve-tone equal temperament system in music theory. In musical terms, changes in pitch are reflected as changes in frequency values within the audio signal. Typically, the frequency ratio between notes an octave apart is 2:1. In the twelve-tone equal temperament system, the frequency ratio between adjacent semitones is the twelfth root of two. Consequently, the horizontal axis of a musical signal changes exponentially, and when represented in three-dimensional space, the pitch changes correspond to a spiraling frequency pattern, as illustrated in Fig. 2 below. This visual representation highlights the frequency changes associated with different pitch levels more intuitively.

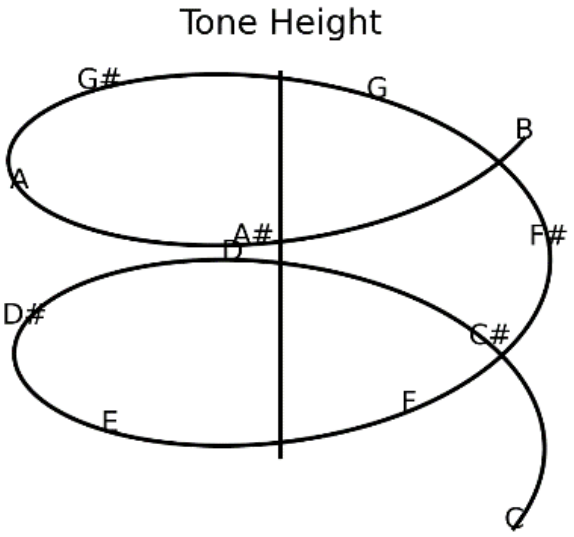


Fig. 2. Three-dimensional representation of sound level.

The distinct advantage of the PCP feature lies in its ability to process the spectral energy of an audio signal alongside its musical features, thereby providing a more accurate representation of the musical characteristics within the audio data. This enhanced representation is particularly beneficial when analyzing music-related audio signals. The following section will expand on the formula used to calculate the PCP feature for a single frame state:

$$p(k) = \lfloor 12 * log_2(k * \frac{f_{sr}}{N} / f_{rel}) \rfloor mod 12$$

where $f_{rel}$ is the reference frequency value of the lowest scale group, the lowest scale group includes the scales C1, D1, E1, F1, G1, Al, B1; $f_{sr}$ is the sampling frequency, $N$ represents the number of sampling points, $f_{sr}/N$ denotes the transform frequency interval of the Fourier Transform, and $k * \frac{f_{sr}}{N}$ denotes the frequency of each component in the frequency domain, so $k * \frac{fsr}{N * f_{rel}}$ represents the The frequency ratio of the component to the level, and all the components corresponding to the frequency value of the same level are summed up according to the above formula to get the twelve-dimensional PCP main melody feature vector:

$$PCP[p] = \sum_{k:p(k)=p} |X[k]|^2 \ , \ p = 1,2,\ldots,12$$

where $X(k)$ is the energy spectrum obtained by Fourier transforming the audio data of the main melody, $k$ is the index of the Fourier transformed component, and $p$ is the ordinal number corresponding to the twelve-tone levels. According to the twelve mean laws in music theory, ignoring the influence of the higher or lower octave, and only considering the frequency

values of the twelve tone levels in the lowest scale group in the music, each component in the frequency domain and the frequency value of the lowest tone level are divided correspondingly to obtain twelve frequency ratios, thus completing the expansion of the components into twelve frequency bands; for all the twelve bands obtained by the components, the components corresponding to the same tone level bands are summed up to get the twelve-dimensional PCP melody feature vector $PCP[p]$ in the whole frequency domain.

From the calculation of the PCP feature in the previous subsection, it is easy to see that it is the spectral information of the music signal is compressed by the frequency rule corresponding to the twelve equal-tempered law, and folded into a twelve-dimensional vector in the form of a tone level profile of the spectrum This calculation process, although the spectral information is endowed with the musical characteristics, does not take into account the possible problems that may exist in the music signal. Usually in musical signals, the notes in the low frequency part are difficult for the human ear to distinguish and hear because of their resonance, so the bass is more blurred and less distinctive in most cases. In addition, there are overtones in common music signals. In the normal human ear mechanism hearing system, overtones do not cause too much interference and influence on the auditory senses, so they are generally not too concerned about overtones. However, in the audio feature representation, when there are too many overtones in the music signal, the process of converting it into a spectrum will occupy more spectral resources, thus affecting the energy of the similar fundamental frequency, generating errors, and affecting the information extraction of the real sound value. Considering that the problems of bass ambiguity and high-frequency overtones are less considered in the current research, this paper introduces a Gaussian filter bank, which is combined with the musical properties of the twelve equal temperament law, to add a window restriction and increase the weights of the fundamental frequencies of the tone levels. The weights of irrelevant frequencies are filtered out, and the mathematical expression of this Gaussian filter is as follows.

$$PCP_{[p]} = \exp\left(-\frac{\left(k^{*}\frac{f_s}{N} - f_{rel}^{*}2^{(O-1)}\right)^2}{2^{*}15^2}\right) * PCP_{[p]}$$

In the above formula, $k * \frac{f_{sr}}{N}$ represents the component frequency value of the sample, and the center frequency is the reference frequency value of the scale corresponding to the octave interval where it is located, $f_{rel} * 2^{(O\cdot1)}$, $o$ represents that the frequency of the sample point at this time corresponds to the frequency range of the $o$ th octave. Because $f_{rel}$ represents the reference frequency value of the lowest scale, when the frequency value of the sampling point is in the frequency range of other octave intervals, because the frequency relationship of different octaves is the 2nd power relationship, for example, the ratio of the frequency values of C2 and C1 is 2, and the ratio of the frequency values of C3 and C1 is 4, so the center frequency becomes the reference frequency value of the original lowest scale group multiplied by an integral multiple of 2, it is now in the octave where the reference frequency value is located. At this point, the frequency values of the twelve semitones within the octave interval become the new center frequencies. These center frequencies are set to correspond with the semitone frequencies in the twelve-tone equal temperament system. This method retains the frequency weights of all notes in this system while filtering out irrelevant frequency values. Consequently, low-frequency noise and high-frequency overtone interference are effectively mitigated, and the fundamental frequencies of the low-frequency band are preserved, addressing the issue of indistinct tone values to some extent.

Fig. 3 illustrates the spectrum of the frequency interval for A4 after Gaussian filtering. It shows that 440Hz has the highest amplitude, indicating it as the center frequency. Other frequencies have amplitudes ranging between 420Hz and 430Hz on the left boundary and between 450Hz and 460Hz on the right boundary. The frequencies of G#4 and B4 fall outside these boundaries, ensuring that effective tone values pass through, demonstrating the efficacy of the filtering. Each Gaussian filter's center frequency corresponds to the twelve semitones between C4 and B4. This filtering method effectively extracts frequency domain energy based on the twelve-tone equal temperament system, mitigating low-frequency noise and high-frequency overtones.

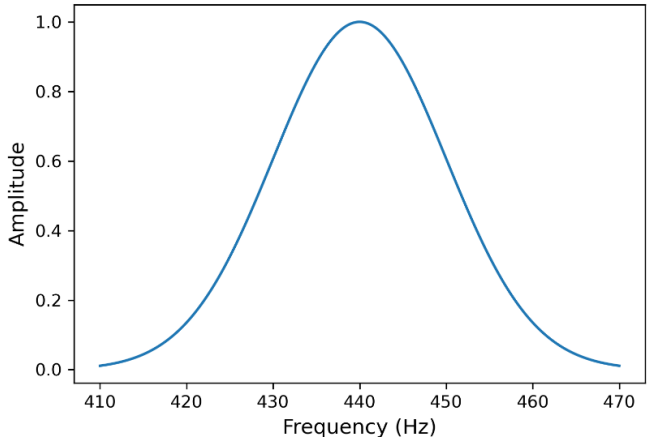


Fig. 3. A4 korst spectral plot of notes.

The primary process in PCP feature calculation involves folding and weighting the spectral energy. In actual music audio, the complex interplay of sounds from various instruments with different pitches, volumes, and rhythms results in significant variability in the chromatic features extracted from each song. This variability introduces multiple levels of complexity, making it challenging to develop a classifier that covers the entire feature space. To manage this, logarithmic compression is used to limit the dynamic range of the features, as detailed by the following mathematical formula:

$$PCP_{Log} = \log\left(1 + \eta * \widehat{PCP}[p]\right)/P[p]_{sum}) \ , \ p = 1,2,\ldots,12$$

$\widehat{PCP}[p]$ is the PCP feature vector obtained after Gaussian filtering as described above, $P[p]_{sum}$ is the sum of all the frequency components corresponding to the twelve semitones, and $\eta$ stands for the compression coefficient, and 100 is used as the compression coefficient in this paper because it has the best performance in the experiment. The ratio of the filtered PCP feature vector to the total frequency components is obtained,

multiplied by the compression coefficient, weighted and summed with 1, and then logarithmically transformed to replace the original PCP feature vector. The above compression method reduces the computation amount of the related feature frequency values, and makes the effect of the features have a better performance ability.

### B. HMM Model

An HMM model is a statistical framework extensively utilized in various natural language processing applications, including speech recognition, automatic lexical annotation, and probabilistic grammar analysis. A process is considered Markovian, or a Markov process, if its future state depends solely on its current state and not on its past states.

$$X(t+1) = f(X(t))$$

Where $X(t)$ denotes the state at time t. Markov processes that are discrete in time and state are called Markov chains.

$$X_n = X(n), n = 0,1,2,\ldots$$

Denotes the results of successive observations of discrete state processes on the time set $T = \{0,1,2,3,\ldots\}$ The result of successive observations of discrete state processes on the time set $T = \{0,1,2,3,\ldots\}$. A Markov chain is a random process that adheres to the following:

The probability distribution of the system's state at time $t+1$ depends only on its state at time t, and is independent of its states prior to t;

The transition from the state at time t to the state at time t+1 is independent of the specific value of t.

A Markov chain model can be defined by the elements (S, P, Q), where:

S is a non-empty set of all possible states of the system, commonly known as the state space. This set can be finite, countable, or any non-empty set. In this paper, S is assumed to be countable, with states denoted by lowercase letters such as i, j etc.

P is the state transition probability matrix of the system, $p_{ij}(k)$ represents the probability of transitioning from state i at time t to state j at time $t+k$. For a Markov chain model in a discrete state space with a finite number of states, the transition probability distribution is expressed as a matrix with $N \times N$ elements, known as the "transition matrix".

$$P_{ij}(t, t+k) = P(q_{t+k} = \theta_j \mid q_t = \theta_i)$$

When $k = 1, P_{ij}(1)$ is called a piece of transfer probability, referred to as transfer probability, and all the transfer probabilities $a_{i_j}$, $1 \le i, j \le N$ can form a state transfer matrix:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}$$

Where $0 \le a_i j \le 1$ and $\sum_j = 1^N a_{ij} = 1$. Q represents the initial probability distribution of the system, with $\pi_i$ indicating the probability of the system being in state $i$ at the initial time.

The Fig. 4 below demonstrates the hidden and observed states using a weather example. In this model, the hidden state (actual weather) is represented by a first-order Markov process, where each state is interconnected. In addition to the probabilistic relationships defined by the Markov process, there is a confusion matrix that outlines the probabilities of the observed states for each corresponding hidden state.

For the weather example, the confusion matrix is shown in Table I.

TABLE I. CONFUSION MATRIX OF WEATHER

| Observed weather | Hide Weather | | | |
|---|---|---|---|---|
| | Dry | Dryer | Wet | Soggy |
| Sunny | 0.6 | 0.2 | 0.15 | 0.05 |
| Cloudy | 0.25 | 0.25 | 0.25 | 0.25 |
| Raining | 0.05 | 0.1 | 0.35 | 0.5 |

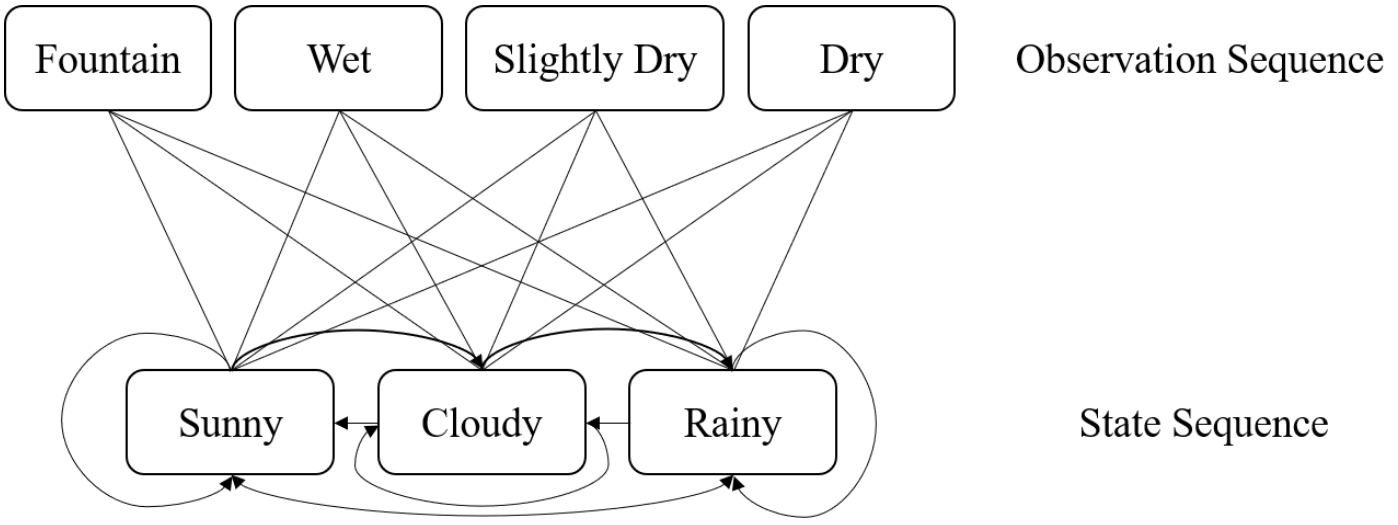


Fig. 4. First-order hidden markov processes.

A Hidden Markov Model (HMM) is characterized by five elements: two sets of states and three probability matrices. The hidden states $S_4$ satisfy the Markov property and represent the underlying processes that are not directly observable ($S_1, S_2, S_3$, etc.). The observable states (OOO) correspond to these hidden states and can be directly measured ( $O_1, O_2, O_3$ , etc). It's important to note that the number of observable states doesn't necessarily match the number of hidden states.

The initial state probability distribution, π, defines the probabilities of the system starting in each hidden state at t=1, $P(S_1) = P_1, P(S_2) = P_2, P(S_3) = P_3$ , describes the probabilities of transitioning from one hidden state to another. Additionally, the observation probability matrix B —often called the confusion matrix—provides the probabilities of each observable state given a hidden state. This comprehensive framework allows HMMs to model complex sequences where the true states are not directly observable but can be inferred through observed data.

$$A = \begin{bmatrix} a_{11} & \dots & a_{1N} \\ \vdots & & \vdots \\ a_{N1} & \dots & a_{NN} \end{bmatrix}$$

$a_{ij}$ denotes the probability that the state is $S_i$ at time t and $S_j$ at time $t + i$. A describes the transfer probabilities between states in the Hidden Markov Model. The observed state transfer probability matrix B (Confusion Matrix) is as follows:

$$B = \begin{bmatrix} b_{11} & \dots & b_{1N} \\ \vdots & & \vdots \\ b_{M1} & \dots & b_{MN} \end{bmatrix}$$

### C. Application of Hidden Markov Models

Once a system is defined as a Hidden Markov Model (HMM), it can solve three fundamental problems. The first two are pattern recognition tasks: calculating the probability of a specific observation sequence given the HMM, and determining the most likely sequence of hidden states that could produce the observed sequence. The third problem is to generate an HMM from a given sequence of observations.

*1) Evaluation:* This involves assessing which of several Hidden Markov Models (represented by sets of $\Pi$, A, B) is most likely to have generated a specific observation sequence. For instance, we might have different HMMs for "summer" and "winter" based on seasonal variations in seaweed humidity. By evaluating the probability of observed humidity sequences, we can determine the most appropriate model, thus inferring the current season. In speech recognition, this method is used to identify words by comparing the observation sequences against multiple HMMs, each representing a different word. The forward algorithm is employed to compute the probability of the observation sequence for each HMM, enabling the selection of the most likely model.

*2) Decoding:* This task involves finding the most probable sequence of hidden states that could generate a given sequence of observations. This is particularly valuable as hidden states often represent significant, unobservable information. For example, consider a scenario where a blind hermit can observe the state of seaweed but wants to infer the underlying weather conditions (the hidden states). The Viterbi algorithm is used in such cases to determine the most likely sequence of hidden states given the observed data, providing insights into the unobservable processes.

*3) Learning:* The third and most challenging problem in HMMs is generating an appropriate Hidden Markov Model from a sequence of observations. This involves estimating the optimal HMM parameters—$\Pi$, A and B—that best describe the observed sequence and the associated hidden states. This process, known as learning or parameter estimation, is crucial when the transition and observation matrices (A and B) cannot be directly measured, which is often the case in practical applications. The forward-backward algorithm is typically employed for this purpose, as it allows for the iterative refinement of the model parameters to maximize the likelihood of the observed data given the model.

### D. Implementation of Hidden Markov Models

Hidden Markov models, described by a vector and two matrices ($\Pi$, A, B), are of great value for real systems, and although they are often only an approximation, they are robust to analysis. Hidden Markov models typically solve problems such as: evaluation, decoding, and learning.

We use a forward algorithm to compute the probability of a T -long sequence of observations:

$$Y^{(k)} = y_{k_1} y_{k_2}, \dots\dots, y_{k_{T-1}} y_{k_T}$$

To compute the probability of an observation sequence of length T, the forward algorithm is employed. This method involves recursively determining the probability of the observation sequence for a given HMM. We start by defining the partial probability, which represents the likelihood of reaching an intermediate state within the sequence. Then, we describe how to compute these local probabilities at t=1 and for subsequent times t=n (where n>1). The following Fig. 5 illustrates the weather states and the first-order state transitions for observation sequences labeled as dry, wet, and soaked conditions:

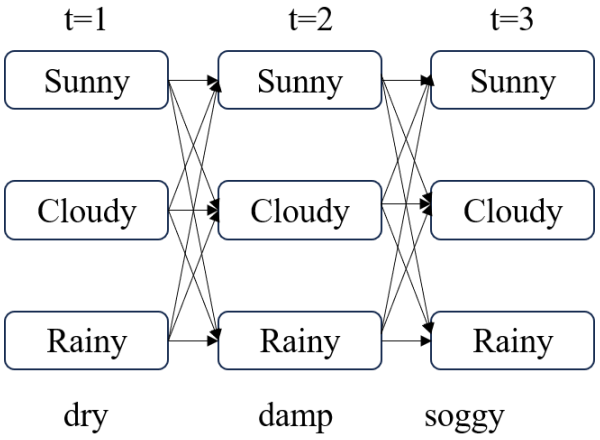


Fig. 5. First order state transfer diagram.

We define the local probability of being in state j at time t as $\alpha_t(j)$. This local probability is computed using the formula:

$$\alpha_i(j) = \Pr(\text{Observe state} \mid \text{Hidden state j}) \times \Pr(\text{all paths to state j at time t})$$

For the final observed states, this local probability includes the likelihood of reaching these states through all possible paths in the lattice. By summing these final local probabilities, we obtain the total probability of the observed sequence given the Hidden Markov Model (HMM).

To compute the local probability $\alpha_t(j)$ at t = 1, we use the initial probabilities, since there are no prior paths. Thus, the probability of being in the current state at t = 1 is the initial probability, represented as Pr (state t=1) =P(state). Consequently, the local probability at t=1 is calculated by multiplying the initial probability of being in the current state with the corresponding observation probability:

$$\alpha_1(j) = \pi(j) \times b_{jk_1}$$

Where $\pi(j)$ is the initial probability of state $j$, and $b_{jk_1}$ is the probability of observing the initial observation given state j. So, the local probability of state j at the initial moment depends on the initial probability of this state and the observation probability that we have seen at the corresponding moment.

See the observation probability at the corresponding moment. Calculate the local probability for t>1:

$$\alpha_1(j) = \pi(j)^* b_{jk_1}$$

We can assume, recursively, that the probability of the observed state given the hidden state Pr (Observe state | Hidden state j) is already known. Now, we focus on the probability of all paths leading to state j at time t (Pr (all paths to state j at time t) ). The number of paths required to compute α_ι (j) increases exponentially with the sequence of observations, but at moment t-1 $\alpha_{t-1(j)}$ gives the probability of all previous paths to this state, so we can define $\alpha_t(j)$ at moment t by the local probability at moment t-1:

$$\alpha_{t+1}(j) = b_{jk_{t+1}} \sum_{i=1}^{n} \alpha_t(j) a_{ij}$$

Therefore, this probability we compute is equal to the sum of the corresponding observation probability (i.e., the probability of observing a symbol in state j at time t+1) and the probability of arriving at this state at that moment, from the product of the result of the computation of each localized probability from the previous step and the corresponding state-transfer probability multiplied by the product of the corresponding state-transfer probabilities, as shown in Fig. 6.

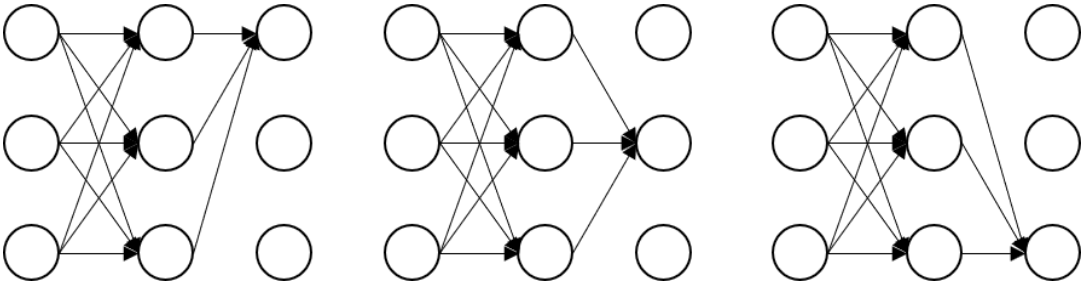

Fig. 6. Final localized probability transfer map.

Now we can recursively compute the probability of a sequence of observations after a given Hidden Markov Model (HMM). We start by computing $\alpha_2(j)$ at t=2 from the local probability $\alpha_i(j)$ at t=1, $\alpha_3(j)$ at t=3 from $\alpha_2(j)$ at t=2, and continue this process until t=T. The probability of the entire observation sequence for a given HMM is the sum of the local probabilities at t=T:

$$\Pr(Y^{(k)}) = \sum_{j=1}^{n} \alpha_T(j)$$

To efficiently compute the probability of an observation sequence given an HMM, we use the forward algorithm. This algorithm employs recursion to avoid the exhaustive computation of all possible paths in the lattice. Using this approach, we can evaluate multiple HMMs by applying the forward algorithm to each one and then selecting the model that yields the highest probability for the given observation sequence.

For generated observation sequences, the most probable model parameters are determined and optimized using the forward-backward algorithm. The essential problems addressed by HMMs include evaluation (using the forward algorithm) and decoding (using the Viterbi algorithm). The evaluation measures the relative fitness of a model, while decoding infers the sequence of hidden states. Both processes depend on the HMM parameters: the state transition matrix A, the observation matrix B and the initial state probability vector $\Pi$.

In the forward algorithm we define the local probability $\alpha_t(i)$, call it the forward variable:

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, q_t = S_i \mid \lambda)$$

Similarly, we can define a backward variable $\beta_t(i)$:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T \mid q_t = S_i, \lambda)$$

The backward variable represents the probability of a sequence of local observations from the moment t+1 to the termination moment, knowing the Hidden Markov Model $\lambda$ and the fact that ι moments are located in the hidden state $s_i$. Also similar to the forward algorithm, we can compute the backward variable recursively from backward to forward (hence the term backward algorithm): Initially, the backward variable for all states at time $t = T$ is 1

$$\beta_T(i) = 1 \quad 1 \le i \le N$$

Inductively, recursively calculate for each time point, $t = T-1, T-2, \ldots, 1$ at the time of the backward variable:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j) \quad t = T-1, T-2, \ldots, 1 \quad 1 \le i \le N$$

This approach allows for the computation of backward variables for all hidden states at each point in time. To calculate the probability of observing a sequence using the backward algorithm, one needs to sum the backward variables (local probabilities) at t=1. The following Fig. 7 shows the relationship between the backward variables at moment t+1 and at moment t:

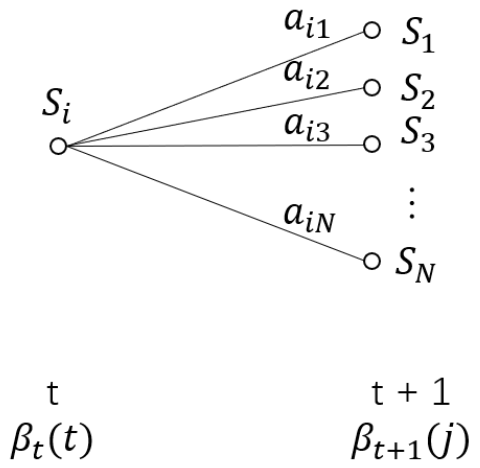


Fig. 7. Forward-backward variable relationships.

Among the three basic problems of Hidden Markov Models (HMM), the third problem of HMM parameter learning is the most difficult, because for a given sequence of observations O, there is no method that can accurately find an optimal set of Hidden Markov Model parameters $(\Pi, A, B)$ to maximize $P(O|\lambda)$. As a result, scholars retreat to the second-best solution, which cannot make $P(O|\lambda)$ globally optimal, and seek for a solution that makes it locally optimal, and the forward-backward algorithm becomes an alternative solution to the Hidden Markov Model learning problem. We first define two variables. Given an observation sequence O and a Hidden Markov Model $\lambda$, define the probability variable of being in the hidden state $S_i$ at time t as:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda)$$

Regarding the definition of the forward variable $\alpha_t(i)$ and the backward variable $\beta_i(i)$, we can easily express the above equation in terms of forward and backward variables as:

$$\gamma_t(i) = \frac{\alpha_t(i)\, \beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\, \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\, \beta_t(i)}$$

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

## IV. Marching Band Performance Recognition System Design

The automatic music recognition system developed in this paper comprises two primary components: the music feature extraction module, which utilizes the enhanced PCP feature extraction method previously described, and the modeling module. The modeling module involves gathering modeled music labels and conducting the training and prediction phases of the model, as shown in Fig. 8.

The automatic music recognition system presented in this paper is divided into two primary sections. The music feature extraction module, shown in the dotted box on the left, utilizes the improved PCP features discussed before. The model module, depicted in the dashed box on the right, focuses on the HMM model and the creation of automated music tags. The details of the model module will be further explained in the following sections.

First, the user uploads audio and transmits it to the back-end via Axios' XMLHttp Request. send method; the back-end receives the request and begins to reason about the audio through format detection and returns the inference results; the front-end displays the inference results and can synchronize with the results of playing the audio; the user can choose to download the pipe using the VUE download File method; the orchestra plays the music or re-uploads the music; the user can choose to download the pipe using the VUE download File method. Users can choose to download the music played by the orchestra or re-upload the music by using the VUE download File method. The processing state is shown in Fig. 9.

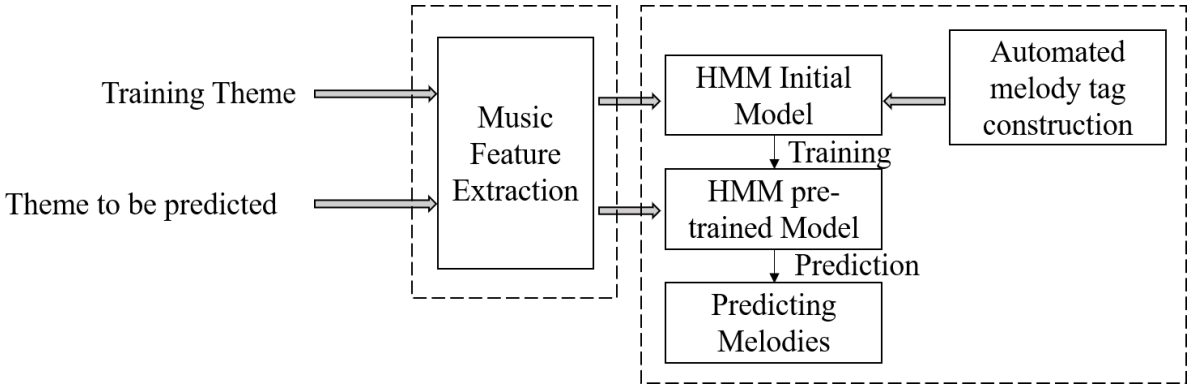


Fig. 8. Framework diagram of automatic music recognition system.

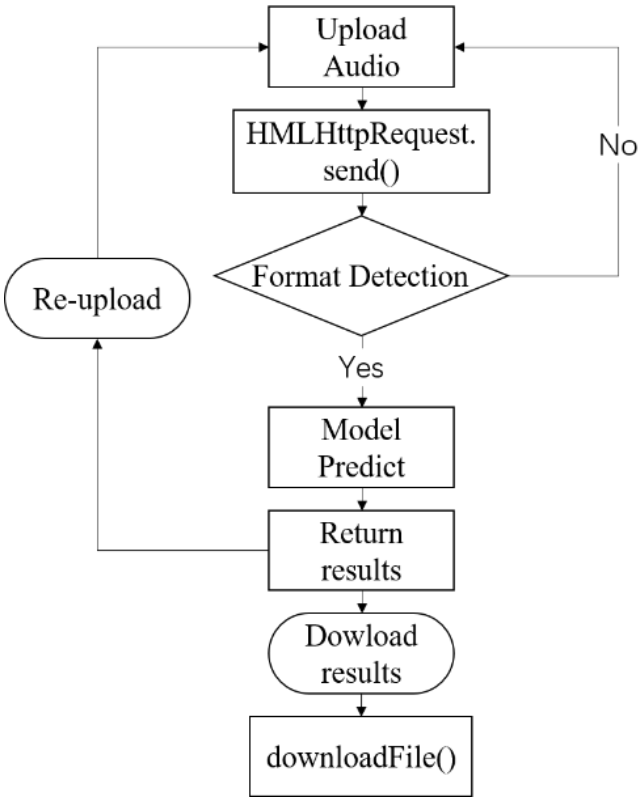


Fig. 9. Platform process.

## V. Results and Discussion

### A. Experimental Data Collection

The source data for this paper consists of 455 MIDI music files obtained from an online MIDI music library. Of these, 450 files are used for training and 5 for testing. The datasets are preprocessed to extract the main melody and accompaniment tracks. The main melody is converted to WAV format for feature extraction, while the accompaniment is labeled using the proposed automatic music label construction method. Both data types are named consistently for model training purposes.

The improved PCP feature vector for the main melody, as proposed in this paper, is utilized for model feature extraction and serves as the observation vector for the HMM. To assess the robustness of the system, experiments were conducted to evaluate its performance on real-world audio recordings with background noise and microphone variations. The results indicated that while the system performed well under controlled conditions, its accuracy decreased in the presence of significant noise and variations, suggesting the need for further refinement and noise reduction techniques. The HMM consists of six states, excluding the initial and termination states. Each active state employs a single Gaussian observation function with a diagonal matrix, an average vector, and a change vector. After training the model, five files are randomly selected from the test dataset. The improved PCP feature vectors are extracted and inputted into the model for wind music prediction, and the predicted sequences are recorded. These steps are then repeated using traditional PCP features as observation vectors for comparison purposes.

To evaluate the accuracy of the predicted wind music, the results are compared against the correct harmonic sequences determined by professional music researchers using established music theory. The accuracy of the system's recognition is mathematically defined as follows:

$$P_{true} = \frac{N_{sum} - N_{false}}{N_{sum}}$$

In the above formula, $N_{sum}$ represents the total number of accompanied piped tunes of a single tested music file, and $N_{false}$ represents the number of incorrectly recognized piped tunes, and their difference represents the meaning, which is equal to the number of piped tunes that appear to be different in the results of all the piped tunes generated by the system recognition of the tested music file in this paper, compared to the results of all the piped tunes obtained by manual recognition.

### B. Tests Results

Statistics of the correctness of the five test music files recognized by the system for wind music tunes were obtained, and the data of the experimental results are shown in the following Table II.

TABLE II. Comparison of Traditional PCP and Modified PCP Results

| Training data | Test data (piece name) | System Type | Recognition Accuracy |
|---|---|---|---|
| Music files in the MIDI music library | Liberty Bell March | Legacy PCP+HMM | 79.32 |
| | | Revised PCP+HMM | 84.77 |
| | British Grenadiers March | Legacy PCP+HMM | 76.41 |
| | | Revised PCP+HMM | 82.52 |
| | El Capitan | Legacy PCP+HMM | 72.76 |
| | | Revised PCP+HMM | 75.22 |
| | Entry of the Gladiators | Legacy PCP+HMM | 78.01 |
| | | Revised PCP+HMM | 84.29 |
| | The Thundered | Legacy PCP+HMM | 72.36 |
| | | Revised PCP+HMM | 74.47 |

The data presented in the table indicates that the improved PCP features used in this study enhance the accuracy of wind music recognition compared to traditional PCP features. Specifically, the experimental results show that the improved PCP features increase the recognition accuracy for the pieces "Liberty Bell March," "British Grenadiers March," and "Entry of the Gladiators" by 5.25%, 6.11%, and 6.28%, respectively. For the pieces "El Capitan" and "The Thundered," the recognition accuracy improved by 2.46% and 2.11%, respectively. Overall, the improved PCP features proposed in this study provide better recognition performance for wind music than the traditional PCP features.

Additionally, to comprehensively evaluate the performance of the wind music recognition system designed in this paper, a traditional template matching model was used for control analysis. This comparison helps to further validate the effectiveness of the improved PCP features and the overall recognition system. The experimental results obtained from the final analysis are shown in the following Table III.

TABLE III. COMPARISON OF SYSTEM SYNTHESIS RESULTS

| Training data | Test data (piece name) | System Type | Recognition Accuracy |
|---|---|---|---|
| Music files in the MIDI music library | Liberty Bell March | Legacy PCP+ Template Matching | 74.32 |
| | | Revised PCP+HMM | 84.36 |
| | British Grenadiers March | Legacy PCP+ Template Matching | 72.89 |
| | | Revised PCP+HMM | 82.66 |
| | El Capitan | Legacy PCP+ Template Matching | 69.02 |
| | | Revised PCP+HMM | 74.87 |
| | Entry of the Gladiators | Legacy PCP+ Template Matching | 72.31 |
| | | Revised PCP+HMM | 83.85 |
| | The Thundered | Legacy PCP+ Template Matching | 68.38 |
| | | Revised PCP+HMM | 73.94 |

The data from the table indicates that the HMM model combined with the improved PCP features significantly outperforms the template matching model in terms of pipe music recognition accuracy. Specifically, when comparing the results of the pipe music recognition system using the improved PCP features and the HMM model to those using traditional PCP features and template matching, the recognition accuracy improved by 9.55%, 9.77%, and 11.25% for "Liberty Bell March," "British Grenadiers March," and "Entry of the Gladiators," respectively. For "El Capitan" and "The Thundered," the improvements were 5.35% and 5.56%, respectively. These results demonstrate that the HMM model provides better performance compared to template matching.

However, a closer look at the data reveals that "El Capitan" and "The Thundered" have lower overall recognition rates compared to the other three songs. Neither the improved PCP features nor the use of the HMM model had a substantial impact on these pieces, resulting in relatively low recognition accuracy. To understand the reasons behind this, a further analysis of the accompanying wind music sequences derived from the test set music files by professionals was conducted.

This analysis revealed that the primary reason for the decreased recognition rate in these pieces is related to the complexity of the wind music analysis. Specifically, the wind music involved in this study often contains repeated sound values, which are particularly challenging to recognize accurately. For example, both the F major wind piece [F, A, C] and the D minor piece [D, F, A] share similar tonal components, leading to frequent recognition errors. This issue is a significant factor contributing to the lower recognition rates observed for these songs.

As illustrated in Fig. 10 and 11, the first two tone values in F major and the last two-tone values in D minor are identical. This similarity can cause the system to misidentify sections of the main melody, confusing F major with D minor due to their high degree of resemblance, thus affecting the overall recognition accuracy. Similarly, Fig. 12 and 13 show that C major and A minor wind music also share common features, both containing C and E as root notes. This overlap makes it challenging to distinguish between these keys during the recognition process. In the pieces with lower recognition rates, such as those involving C major, A minor, F major, and D minor, these shared tonal characteristics lead to frequent misidentifications, resulting in decreased recognition accuracy compared to other pieces.

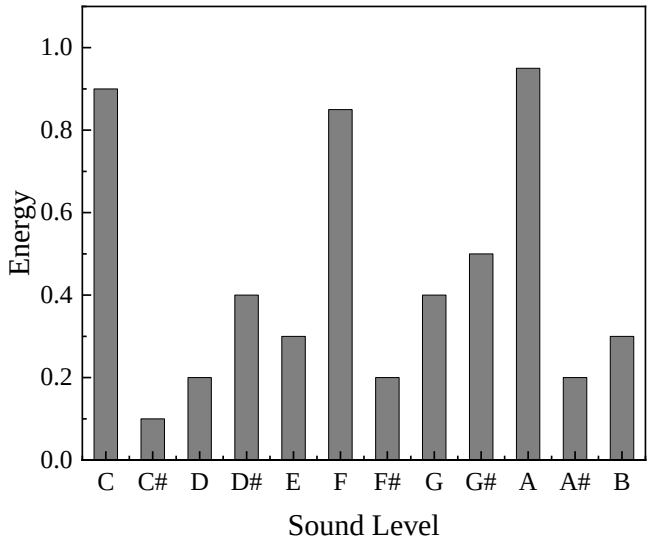


Fig. 10. F-major's PCP feature template.

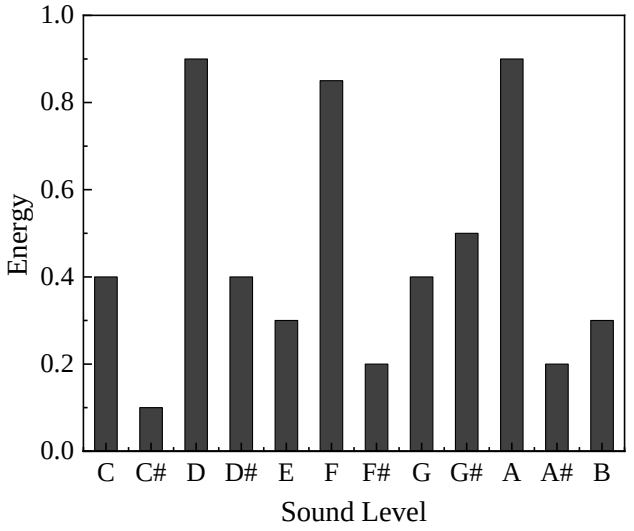


Fig. 11. D-minor's PCP feature template.

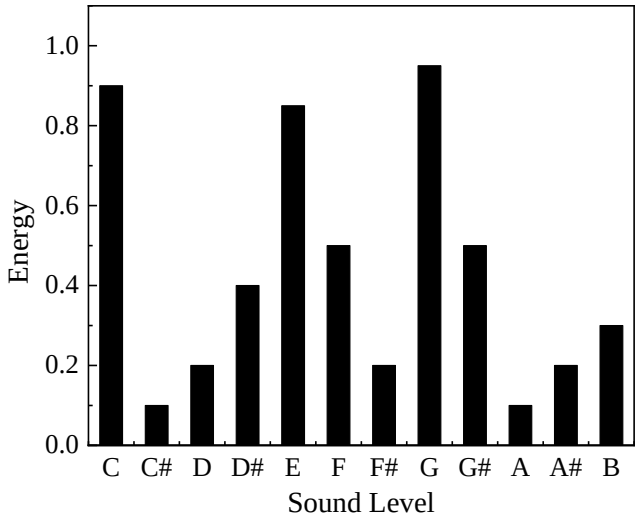


Fig. 12. C-major's PCP feature template.

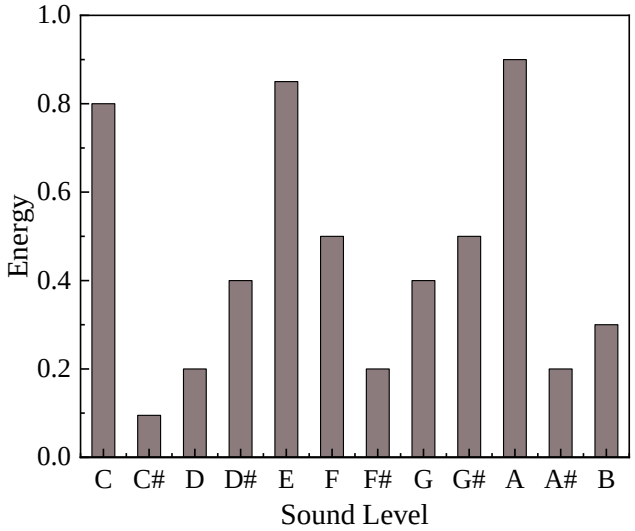


Fig. 13. A-minor's PCP feature template.

## VI. Conclusion

This paper presents a comprehensive study on the automatic recognition of marching band performances, utilizing advancements in music information retrieval and signal processing. The research focused on overcoming the limitations of traditional music feature extraction methods by introducing an improved Pitch Class Profile (PCP) feature extraction technique. By mapping audio signals to musical notes more accurately, the improved PCP features, combined with Hidden Markov Models (HMM), provided a robust framework for recognizing and sequencing marching band performances.

The system developed in this study consisted of two main components: a music feature extraction module and an HMM-based modeling module. The feature extraction module used the improved PCP features to convert audio signals into analyzable data, while the modeling module applied HMMs to decode these features into recognizable performance sequences. The system was trained on a dataset of 450 MIDI music files and tested on an additional five files. Experimental results demonstrated that the improved PCP features significantly enhanced the recognition accuracy compared to traditional PCP features and template matching models. Recognition rates improved by up to 6.28% for various marching band pieces. However, some pieces, such as "El Capitan" and "The Thundered," had lower recognition rates due to the presence of similar tonal values, highlighting the complexity of music recognition. This research contributes to the field of music information retrieval by providing an enhanced feature extraction method and a robust modeling framework. The findings have practical implications for developing automated music recognition systems, which can be applied in music education, digital archiving, and cultural preservation.

Future work should address the challenges of overlapping tonal values by developing more sophisticated feature extraction methods. Additionally, testing the system on real-world audio recordings with background noise and microphone variations will be crucial to enhance its practical applicability. Expanding the dataset to include various musical styles and real-world audio recordings would provide a more comprehensive evaluation of the system's robustness and versatility.

## References

[1] Keller, Robert, et al. "Automating the explanation of jazz chord progressions using idiomatic analysis." Computer Music Journal 37.4 (2013): 54-69.

[2] Qi, Yuting, John William Paisley, and Lawrence Carin. "Music analysis using hidden Markov mixture models." IEEE Transactions on Signal Processing 55.11 (2007): 5209-5224.

[3] Ajmera, Jitendra, Iain McCowan, and Herve Bourlard. "Speech/music segmentation using entropy and dynamism features in a HMM classification framework." Speech communication 40.3 (2003): 351-363.

[4] Vincent, Emmanuel, and Xavier Rodet. "Music transcription with ISA and HMM." Independent Component Analysis and Blind Signal Separation: Fifth International Conference, ICA 2004, Granada, Spain, September 22-24, 2004. Proceedings 5. Springer Berlin Heidelberg, 2004.

[5] Shibata, Go, Ryo Nishikimi, and Kazuyoshi Yoshii. "Music Structure Analysis Based on an LSTM-HSMM Hybrid Model." ISMIR. 2020.

[6] Nishikimi, Ryo, et al. "Audio-to-score singing transcription based on a CRNN-HSMM hybrid model." APSIPA Transactions on Signal and Information Processing 10 (2021): e7.

[7] Calvo-Zaragoza, Jorge, Alejandro H. Toselli, and Enrique Vidal. "Hybrid hidden Markov models and artificial neural networks for handwritten music recognition in mensural notation." Pattern Analysis and Applications 22 (2019): 1573-1584.

[8] Mor, Bhavya, Sunita Garhwal, and Ajay Kumar. "MIMVOGUE: modeling Indian music using a variable order gapped HMM." Multimedia Tools and Applications 80 (2021): 14853-14866.

[9] Li, Tao, et al. "Music sequence prediction with mixture hidden markov models." 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019.

[10] Qian, Guo. "A music retrieval approach based on hidden markov model." 2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). IEEE, 2019.

[11] Chen, Yanjiao. "Automatic classification and analysis of music multimedia combined with hidden markov model." Advances in Multimedia 2021 (2021): 1-7.

[12] Nishikimi, Ryo, et al. "Bayesian singing transcription based on a hierarchical generative model of keys, musical notes, and f0 trajectories." IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020): 1678-1691.

[13] Ntalampiras, Stavros, and Ilyas Potamitis. "A statistical inference framework for understanding music-related brain activity." IEEE Journal of Selected Topics in Signal Processing 13.2 (2019): 275-284.

[14] Uehara, Yui, Eita Nakamura, and Satoshi Tojo. "Chord function identification with modulation detection based on HMM." Perception, Representations, Image, Sound, Music: 14th International Symposium, CMMR 2019, Marseille, France, October 14–18, 2019, Revised Selected Papers 14. Springer International Publishing, 2021.

[15] Mor, Bhavya, Sunita Garhwal, and Ajay Kumar. "A systematic literature review on computational musicology." Archives of Computational Methods in Engineering 27 (2020): 923-937.

[16] Shibata, Kentaro, et al. "Joint transcription of lead, bass, and rhythm guitars based on a factorial hidden semi-Markov model." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

[17] Wang, Changhong, et al. "HMM-based glissando detection for recordings of Chinese bamboo flute." (2019).

[18] Nápoles López, Néstor, Claire Arthur, and Ichiro Fujinaga. "Key-finding based on a hidden Markov model and key profiles." Proceedings of the 6th International Conference on Digital Libraries for Musicology. 2019.

[19] Ens, Jeff, and Philippe Pasquier. "Mmm: Exploring conditional multi-track music generation with the transformer." arXiv preprint arXiv:2008.06048 (2020).

[20] Ycart, Adrien, et al. "Blending acoustic and language model predictions for automatic music transcription." (2019).

[21] Mor, Bhavya, Sunita Garhwal, and Ajay Kumar. "A systematic review of hidden Markov models and their applications." Archives of computational methods in engineering 28 (2021): 1429-1448.

[22] Brancatisano, Olivia, Amee Baird, and William Forde Thompson. "A 'music, mind and movement'program for people with dementia: Initial evidence of improved cognition." Frontiers in psychology 10 (2019): 445451.

[23] Hernandez-Olivan, Carlos, and Jose R. Beltran. "Music composition with deep learning: A review." Advances in speech and music technology: computational aspects and applications (2022): 25-50.

[24] Plut, Cale, et al. "PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games." Proceedings of the 17th International Conference on the Foundations of Digital Games. 2022.

[25] Bello J P, Pickens J. A Robust Mid-Level Representation for Harmonic Content in Music Signals[C]//ISMIR. 2005, 5: 304-311.

[26] Foote J T. Content-based retrieval of music and audio[C]//Multimedia storage and archiving systems II. SPIE, 1997, 3229: 138-147.

[27] Logan B. Mel frequency cepstral coefficients for music modeling[C]//Ismir. 2000, 270(1): 11.

[28] Müller M. Information retrieval for music and motion[M]. Heidelberg: Springer, 2007.

[29] Sheh A, Ellis D P W. Chord segmentation and recognition using EM-trained hidden Markov models[J]. 2003.

[30] Wu Y, Carsault T, Yoshii K. Automatic chord estimation based on a frame-wise convolutional recurrent neural network with non-aligned annotations[C]//2019 27th European Signal Processing Conference (EUSIPCO). IEEE, 2019: 1-5.