SalsaAsst: Beat Counting System Empowered by Mobile Devices to Assist Salsa Dancers

Yudi Dong, Jian Liu, Yingying Chen, and Woo Y. Lee
Stevens Institute of Technology, Hoboken, NJ 07030, USA
{ydong6, jliu28, yingying.chen, wlee}@stevens.edu

***Abstract*—Dancing is always challenging especially for beginners who may lack sense of rhythm. Salsa, as a popular style of dancing, is even harder to learn due to its unique overlapped rhythmic patterns made by different Latin instruments (e.g., Clave sticks, Conga drums, Timbale drums) together. In order to dance in synchronization with the Salsa beats, the beginners always need prompts (e.g., beat counting voice) to remind them of the beat timing. The traditional way to generate the Salsa music with beat counting voice prompts requires professional dancers or musicians to count Salsa beats manually, which is only possible in dance studios. Additionally, the existing music beat tracking solutions cannot well capture the Salsa beats due to its intricacy of rhythms. In this work, we propose a mobile device enabled beat counting system, *SalsaAsst*, which can perform rhythm deciphering and fine-grained Salsa beat tracking to assist Salsa dancers with beat counting voice/vibration prompts. The proposed system can be used conveniently in many scenarios, which can not only help Salsa beginners make accelerated learning progress during practice at home but also significantly reduce professional dancers' errors during their live performance. The developed Salsa beat counting algorithm has the capability to track beats accurately in both real-time and offline manners. Our extensive tests using 40 Salsa songs under 8 evaluation metrics demonstrate that *SalsaAsst* can accurately track the beats of Salsa music and achieve much better performance comparing to existing beat tracking approaches.**

I. Introduction

Dancing is not only a performance art showed on the stage by professional dancers, but a normal physical exercise/therapy and social activity for the general public, which plays an important social role nowadays. In every human culture, people usually have some kinds of music with a regular beat that elicits synchronised body movement (i.e., dancing). However, dancing is always difficult especially for beginners, due to their self-consciousness or lacking sense of rhythm. Importantly, beat perception and synchronisation are mentally demanding for people who, unlike musicians, have little prior experience in beat or rhythm detection [1].

Salsa is a popular form of social dances with origins in Cuba, a major crossroad of Spanish (European) and African cultures. Different from other types of dances, finding beats in Salsa music is even tougher due to the intricacies of Salsa music. Unlike other types of dance music which usually have one dominant rhythm (e.g., bass or drum), Salsa music has many overlapped rhythmic patterns provided by different Latin instruments (e.g., Clave sticks, Conga drums, Timbale drums) playing at the same time. Hence, it is challenging to find Salsa beats from these overlapped rhythmic patterns just by our ears. In order to dance in synchronisation with the Salsa beats, Salsa dancers, especially for beginners, always need the beat counting voice to remind them of the timings of beats. However, the traditional way to generate the Salsa music with beat counting voice requires professional dancers or musicians to count the beats manually and synthesize the beat counting voice with the Salsa music, which is inefficient and not able to be applied in large-scale Salsa songs. Therefore, a system that can automatically count Salsa beats without requiring professional dancers or musicians is highly desirable.

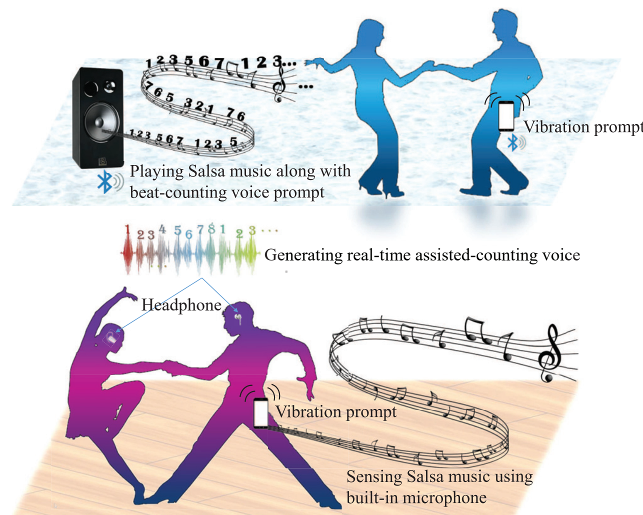


Fig. 1. Illustration on how SalsaAsst assists Salsa dancers.

Some previous studies [2], [3], [4], [5], [6] have already been conducted on tracking beats automatically by analyzing music acoustic signals. However, because of the irregular Salsa rhythmic patterns made by different Latin instruments simultaneously, the beats these approaches detected are usually not at the correct metrical level of Salsa beats (i.e., 1, 2, 3, 4, 5, 6, 7, 8). For instance, the metrical levels of these approaches detected may be at half metrical level of Salsa beats (i.e., 1, 3, 5, 7 or 2, 4, 6, 8) or at twice metrical levels of Salsa beats (i.e., 1, 1.5, 2, 2.5,..., 8, 8.5). That indicates that these approaches would lead to a mass of over/under detections for tracking Salsa beats. In addition, another direction is using the motion sensing device (e.g., Motion Analysis Falcon [7] or Kinect [8]) to obtain the 3-D motion of a dance master and analyze the

2155-6814/17 $31.00 © 2017 IEEE
DOI 10.1109/MASS.2017.25

motion data to get the timings of moving his/her steps, and then remind users of these timings using vibration or counting voice. However, these approaches have to involve professional dancers and need dedicated motion sensors, which restricts its usage scenarios and requires additional cost.

To address these issues, we implement a mobile device based beat counting system, *SalsaAsst*, to assist Salsa dancers with assisted-counting voice/vibration prompts via mobile devices (e.g., smartphones). The proposed *SalsaAsst* can perform rhythm deciphering and fine-grained beat tracking, not only helping Salsa beginners get accelerated learning process during practice at home and be more confident on dance floor but also eliminating professional dancers' errors during their live performance. Figure 1 is an illustration on how *SalsaAsst* assists dancers during their practice sessions or on the stage. Specifically, *SalsaAsst* is able to process Salsa music files offline and generate a new music file which has both the original Salsa music and its synchronous beat counting voice. With the help of the new music file which can be played through the Bluetooth speaker, people may feel much easier to learn Salsa dance. Meanwhile, *SalsaAsst* is also able to sense the Salsa music via the built-in microphone and generate real-time assisted-counting voice through headphones or vibration prompts to remind the dancers of Salsa beat timings.

More specifically, our system only uses Salsa music audio signal for detecting the Salsa beats. The onset detection algorithm fist analyzes the audio signal in frequency domain using Short-time Fourier Transform (STFT) to obtain the spectrogram and then computes Onset Strength Curve (OSC). By locating the local maxima of smoothed OSC, we can obtain the onset positions. To detect Salsa beats, our algorithm first estimates the Salsa beat interval using onset positions, and then track the beats from onsets based on the estimated Salsa beat interval. We develop two alternative modes, *Offline Beat Counting Mode* and *Online Beat Counting Mode*, in our system for flexible usage. The offline mode can process any Salsa dance song files to generate the beat counting voice/vibration prompts along with the original Salsa music. The online mode leverages the built-in microphone on mobile devices to receive the acoustic signal of Salsa music and generate the real-time prompts (i.e., beat counting voice and vibration).

The main contributions of our work are summarized as follows:

- Our proposed *SalsaAsst* can perform rhythm deciphering and fine-grained beat tracking for Salsa music signal and automatically count Salsa beats to remind the dancers of the timings of beats without any additional human task or dedicated sensors deployment.
- We show that it is feasible to enable the mobile device to become a virtual assistant for Salsa dance only using the built-in microphone, speakers, and vibrator of the device.
- We develop a novel beat counting algorithm for Salsa music, which has the capability to track Salsa beats accurately in both real-time and offline manners.
- Extensive tests using 40 Salsa dance songs under 8 evaluation metrics show that our system can achieve much better beat tracking performance comparing to the existing beat tracking solutions.

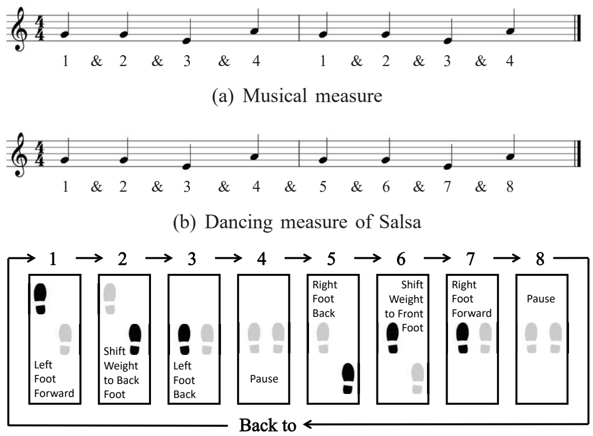
The rest of this paper is organized as follows: Section II discusses related work. Section III introduces some basic knowledge about Salsa and the overview of our proposed system. Section IV and Section V describe *Offline Beat Counting Mode* and *Online Beat Counting Mode* that are core components of our system, respectively. In Section VI, we evaluate the performance of the system. And in Section VII, we perform discussion on the proposed *SalsaAsst* system. Finally, we conclude the paper in Section VIII.

II. Related Work

The crux for assisting in dancing Salsa is to help people to recognize the beats or the timings of moving dance steps. In general, the existing solutions used to assist dancers to recognize the beat timings can be categorized into two groups: (1) dance master relied; and (2) dance music based.

The traditional way of learning Salsa is to attend a dance lesson in which a dance master can count along with the beats of dancing music for the beginners who have difficulty with recognizing the beat timings. In addition, some approaches use dedicated sensors to capture the movements of a professional dancer to get the correct timings of moving steps. For example, Nakamura *et al.* [7] propose a dance training system which captures 3-D motion of a dance master and analyzes the motion to get the timings of moving his/her steps. In order to follow the proper dance steps while dancing, the users need to wear active vibro-devices which can generate vibration to remind users of the step timings. In addition, Misato *et al.* [8] present a system that uses both acoustic features of dance music and skeleton features from movements of a dance master to track the beats. However, these approaches require motion sensing device (e.g., Motion Analysis Falcon [7], Kinect [8]) for capturing the movements of professional dancers, which is costly and not always available.

Furthermore, there are a few existing solutions automatically tracking dance beats through acoustic signal processing of dancing music. For instance, they include beat-template method [2], Bayesian framework [3], dynamic programming [4], [5], and two-state model [6]. In particular, G. Peeters [2] uses beat-template training to estimated musical tempo. A.T. Cemgil [3] formulates tempo tracking in a Bayesian framework. The dynamic programming approach is utilized in [4], [5] to handle beat tracking in the audio signal. M. Davies and M. Plumbley [6] propose a two-state model, which includes general state and context-dependent state, to find the beat positions from music signal. In addition, the work in [9] is similar to ours which also can recognize and count beats by processing audio signal of dancing music. However, these approaches are not applicable for Salsa beat tracking. Different from other types of music, Salsa's rhythmic made by Latin instruments may appear at every two Salsa beats or



(a) Musical measure

(b) Dancing measure of Salsa

(c) Salsa beats corresponding to Salsa steps

Fig. 2. Illustration of the relationship between Salsa musical measure, dancing measure and its corresponding Salsa steps.

half Salsa beat. Therefore, these approaches would lead to a mass of mistakenly detected beats for Salsa music.

Different from the previous work, *SalsaAsst* can accurately track the beats at the correct metrical level embedded in Salsa music by exploiting the proposed *Offline Beat Counting* and *Online Beat Counting* algorithm. Moreover, *SalsaAsst* leverages mobile devices (e.g., smartphone) to raise beat counting voice/vibration prompts to remind users of the timings of Salsa beats. Without any additional human task or sensor deployment, *SalsaAsst* can be used conveniently and flexibly in any scenarios (e.g., live performance, practicing at home or dancing studio) to assist Salsa dancers.

III. Preliminaries and System Overview

In this section, we will introduce the preliminaries of Salsa related measures and challenges of recognizing Salsa beats, then we will describe the system overview of *SalsaAsst*.

A. Preliminaries

Originating in Cuba, Salsa dancing is a representation of a traditional Latin culture. Salsa music and dance have developed since the 1950's from the migrations of Latin people initially to New York and later to other cities in the United States [10]. Since then, Salsa dancing has spread throughout so many parts of the globe and becomes the biggest international dance craze [11].

Salsa music is arranged in bars. It is written in $4/4$ time signature that has four beats in every bar as illustrated in Figure 2(a). A bar (or measure) is a segment of time corresponding to a specific number of beats [12]. For Salsa dancing, it is danced with cycles of two bars making a total of eight beats as shown in Figure 2(b). More importantly, among these eight beats, dancers only need to step on six beats (i.e., 1, 2, 3, 5, 6, 7) as shown in Figure 2(c). The beats "4" and "8" are used for pause.

One of the most challenging parts for Salsa dancing is recognizing these Salsa beats correctly. Different from other

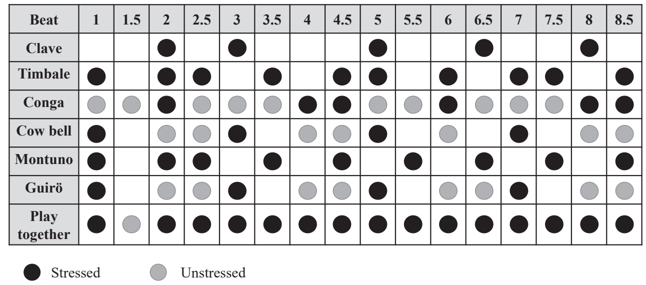


Fig. 3. Rhythmic patterns of different Latin instruments in Salsa music.

kinds of music, the beats of Salsa come from various Latin instruments including Clave sticks, Conga drums, Timbale drums, Montuno, Cow bell, Guir, etc. Every instrument is played with unique beats. Some of them are even played at specific half beats. For instance, Claves is played at 2, 3, 5, 6.5 and 8 every two bars in Salsa songs. Figure 3 shows an example that all these instruments playing together makes it difficult to decipher beats from the Salsa music. Since Salsa music has a bunch of different combinations of these Latin instruments, the rhythm of Salsa is hybrid and complex making it hard to perceive or pinpoint by ears. Even for the professional dancers, it is also possible to have error judgments for Salsa beats during live performance. In order to help Salsa beginners to accelerate their learning process and be more confident on the stage, we propose *SalsaAsst* which can accurately identify the beats of any Salsa music and generate real-time beat counting voice/vibration prompts (i.e., beats 1, 2, 3, 5, 6, 7) to remind of right timings of moving his/her feet while dancing.

B. System Overview

The basic idea of our system is to track Salsa beats from Salsa music signal and generate beat counting voice/vibration prompts via mobile devices to assist Salsa dancers. As illustrated in Figure 4, our system provides two alternative modes, *Offline Beat Counting Mode* and *Online Beat Counting Mode* for flexible usage. The *Offline Beat Counting Mode* of our system can automatically append beat counting voice/vibration prompts to any Salsa music file without any human efforts. Besides, *Online Beat Counting Mode* is designed for generating real-time beat counting voice/vibration prompts for the playing Salsa music sensed by the device's built-in microphone.

Specifically, in the *Offline Beat Counting Mode*, dancers can select a Salsa music file (e.g., MP3 file or WAV file) in their mobile devices. Our system could read the digital audio signal in this music file as input. The data is then processed to mono signal if the input is stereo. After that, the input is fed into *Onset Detection* to obtain the onset positions. Based on the onset positions, our system can estimate Salsa beat interval(SBI). We then develop a method, *Offline Beat Counting*, to detect the beats from onsets by using estimated Salsa beat interval(SBI). According to the detected beats positions, our system can insert beat counting voice and vibration prompts

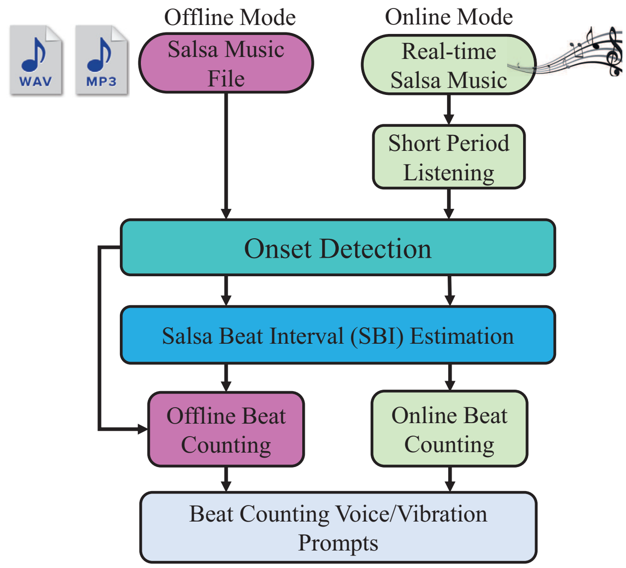


Fig. 4. System overview of SalsaAsst.

corresponding to the Salsa beats. In the *Online Beat Counting Mode*, our system leverages the microphone on mobile devices to receive the acoustic signal of Salsa music and generate the real-time prompts (i.e., beat counting voice and vibration). More specifically, our system listens for a short period of time (i.e., 10 seconds) at the beginning for computing the estimated value of Salsa Beat Interval (SBI). Based on the estimated SBI, our system is able to generate the real-time beat counting voice/vibration prompts along with the dance music to indicate the timings of beats. We leave the details of *Offline Beat Counting Mode* and *Online Beat Counting Mode* to Section IV and Section V, respectively.

IV. Offline Beat Counting Mode

In this section, we first describe *Onset Detection*, and then present *Salsa Beat Interval (SBI) Estimation* based on the onset positions. We finally describe *Offline Beat Counting* algorithm to show how to accurately detect the Salsa beats from onsets.

A. Onset Detection

In this step, we detect all the onsets in the Salsa music signal. An onset refers to the beginning of a sudden burst of energy caused by playing instruments [13]. For example, Figure 5 shows the time domain waveform of playing the Clave in two bars and the corresponding onset positions, which illustrates the relationships between beats and onsets. We can find that the onset is the position where the beat may appear. Therefore, onset detection is the basic step for the further beat tracking.

The procedure of onset detection is to compute the spectrogram from the audio signal, then obtain the Onset Strength Curve (OSC), and find the local maxima from smoothed OSC as onset position. The following are the details.

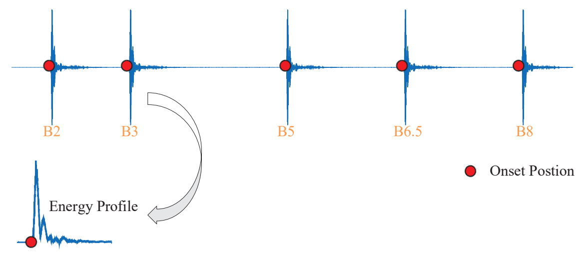


Fig. 5. Onset positions of playing Clave in two bars.

1) Spectrogram: Spectrogram describes the audio signal in terms of energy spread over its frequency components at different times. In *SalsaAsst*, the spectrogram is obtained by computing the log-magnitude of the Short-time Fourier transform (STFT) of audio signal. Consider a length-N music signal $x[n]$ on $0 \leqslant n \leqslant N-1$. The Short-time Fourier transform (STFT) of $x[n]$ is defined as

$$STFT\{x[n]\}(m,k) = \sum_{n=0}^{N-1} x[n]\,\omega[n-mR]\,e^{-j\frac{2\pi}{N}kN}, \tag{1}$$

where m is the index of window frame, R denotes the hop size of STFT which is equal to 32 samples and k represents the kth frequency bin. $\omega[n]$ denotes the window function. Here, we compute the STFT of Salsa music signal with Hamming window function which is defined as

$$\omega[n] = \begin{cases} 0.51 - 0.46cos\left(\frac{2\pi n}{M}\right) & , 0 \leq n \leq M \\ 0 & , \text{Otherwise} \end{cases}, \tag{2}$$

where M is the length of window frame which is equal to 256 samples and n denotes the samples in window. By the way, we use $44.1kHz$ as the sampling rate in STFT.

After computing the STFT, the spectrogram of the music signal $x[n]$ can be obtained by calculating the log-magnitude of the STFT, which defined as

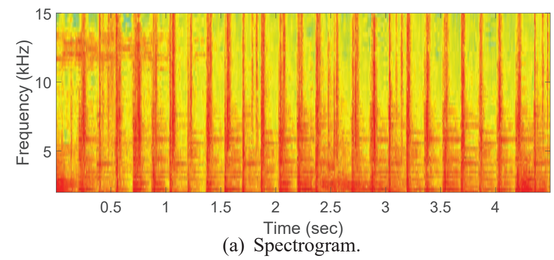
$$Specg\{x[n]\}(m,k) = log_{10}(|STFT\{x[n]\}(m,k)|). \tag{3}$$

As shown in Figure 6(a), the red regions of spectrogram are the frequency regions with a high degree of energy. On the other contrary, the tint regions are the frequency regions with a low degree of energy.

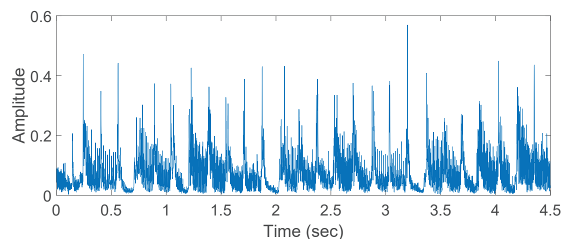
2) Onset Strength Curve: While spectrogram represents the energy distribution of different frequency, the Onset Strength Curve (OSC) is the measurement of the energy changes in spectrogram. By summing the positive changes across all frequency bins in spectrogram, we are able to obtain the raw Onset Strength Curve (OSC), as illustrated in Figure 6(b). The OSC is defined as follows[14]:

$$OSC(m) = \sum_{k=1}^{N} H(Specg(m,k) - Specg(m-1,k)), \tag{4}$$

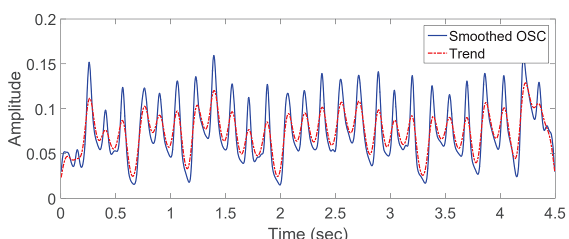
where $Specg(m,k)$ represents the spectrogram of the kth frequency bin of the mth frame. N denotes the number



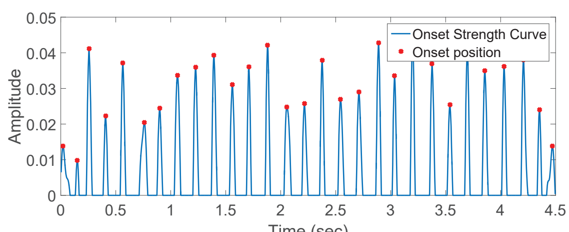
(a) Spectrogram.



(b) Raw Onset Strength Curve (OSC).



(c) Smoothed OSC and its trend.



(d) Detrended OSC and Onset positions.

Fig. 6. Illustration of the procedure of Onset Detection.

of frequency bins. $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function.

3) Onset Positions: After obtaining the raw Onset Strength Curve, the smoothed OSC and its trend can be obtained by leveraging Gaussian low-pass filter with different parameters [5], as shown in Figure 6(c). In particular, the raw OSC is filtered by a Gaussian filter with a cutoff frequency equal to the sampling rate divided by 20 to obtain the smoothed OSC. Its trend is obtained by filtering the smoothed OSC using another Gaussian filter with a cutoff frequency equal to the sampling rate divided by 150. Furthermore, we can compute the detrended OSC by substracting the trend from the smoothed OSC, and find its corresponding peaks as the onset positions, as illustrated in Figure 6(d).

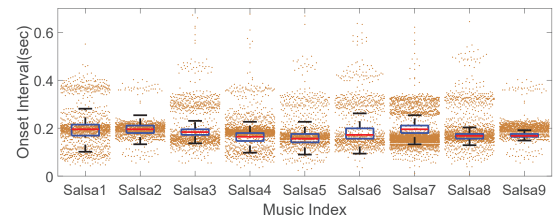


Fig. 7. Dispersions of OIs in nine Salsa songs.

TABLE I
FREQUENCY TABLE OF OIS IN SALSA2

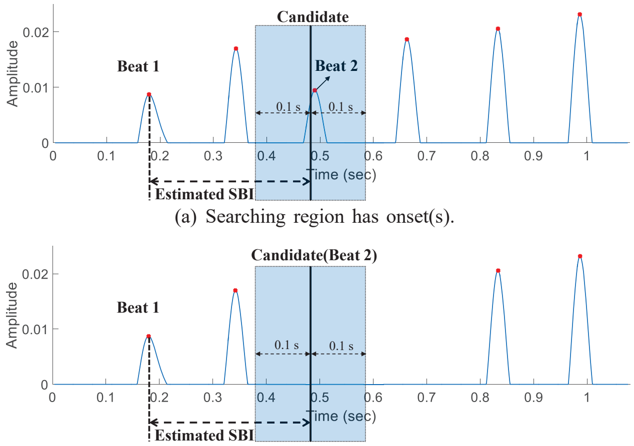
OI(sec)	Frequency	Proportion(%)
0.1640625	330	17.83783784
0.16796875	328	17.72972973
0.171875	221	11.94594595
0.16015625	198	10.7027027
0.17578125	123	6.648648649
0.15625	87	4.702702703
0.1796875	82	4.432432432
...		
0.1328125	5	0.27027027
...		
1.3359375	1	0.054054054

B. Salsa Beat Interval (SBI) Estimation

Based on the onset positions, we can estimate Salsa Beat Interval (SBI). SBI denotes the time interval between the adjacent Salsa beats. For example, the time between beat 1 and 2 is one SBI, while the time between 1 and 1.5 is a half of SBI. Moreover, the SBI is a fixed value for a Salsa song [15]. Our method for Salsa Beat Interval (SBI) estimation is based on the observed relationship between onset and Salsa beat in Salsa music. As shown in Figure 3, different Salsa instruments are played with unique rhythms, for example, the Clave is played at 2, 3, 5, 6.5, 8 where the onsets can be observed there. However, while playing all these instruments in Salsa music, we can find that the onsets appear every half Salsa beat. That is to say, the onset interval (OI) is supposed to be equal to half SBI in Salsa music. We demonstrate that this observed relationship between onset and Salsa beat is reasonable through analyzing the distribution of onset intervals(OIs) in Salsa music. As shown in Figure 7, we list the distributions of onset intervals (OIs) in nine Salsa songs. We can find that most of OIs in a Salsa song are concentrated around a value. That is, most of OIs are approximately equivalent. Therefore, to estimate half SBI of a Salsa song (e.g., Salsa2), we do statistics of the frequency of OIs, as shown in Table I, and compute the arithmetic average of OIs whose frequency proportion is high (i.e., $> 5\%$). This mean value is able be regarded as the approximated value of OI (i.e., half SBI). The estimated SBI then can be obtained by doubling the approximated half SBI.

C. Offline Beat Counting

According to the detected onset positions and estimated SBI, Salsa beats would be searched accurately. Here, we use an example as shown in Figure 8 to explain our method for



(a) Searching region has onset(s).

(b) Searching region has no onset.

Fig. 8. Illustration of how to track Salsa beat from onsets.

beat searching. First of all, we find the first onset position as the first beat position and label it as Beat 1. Starting from Beat 1, we add the estimated SBI to get the candidate of the second beat position (i.e., Beat 2). We then search around the candidate to locate the nearest onset. Since the human ear cannot distinguish the delay/advance of the sound less than $0.1s$ [16], the tolerate delay and advance for the detected beat are $0.1s$. Therefore, the searching region is confined to $\pm 0.1s$ around the candidate. If it has onsets in the searching region, as shown in Figure 8(a), we regard the nearest onset position as the beat position and label it as Beat 2. If not, as illustrated in Figure 8(b)), the candidate would be recognized as the beat position. Then, we repeat the steps above to search the next beat position. This method is able to eliminate the impacts of the false and missed onset detections and then localize the Salsa beats accurately. Moreover, the beat labels are cycled every 8 beats. Our system then generates beat counting voice and vibration prompts corresponding to the beat labels that dancers need to step (i.e., 1, 2, 3, 5, 6, 7).

V. Online Beat Counting Mode

In this section, we introduce how to track beats from real-time Salsa music and generate real-time beat counting voice and vibration prompts on mobile devices. Different from *Offline Beat Counting*, we don't have a Salsa song file to estimate Salsa Beat Interval (SBI). Instead, in *Online Beat Counting*, our system first listens for a short period of time (i.e., 10 seconds) of Salsa music to estimate the SBI via *Onset Detection* and *Salsa Beat Interval (SBI) Estimation*, which have been introduced section IV. Based on the estimated SBI from the 10-seconds historical data, our system then uses the built-in microphone of mobile device to sense the real-time Salsa music signal and save the data in the buffer. We use two threads for real-time beat counting. The main thread is used to play the beat counting voice prompts and generate vibration prompts. Another thread is for searching Salsa beats of the received signal in the buffer. The algorithm of *Online Beat Counting* is provided in Algorithm 1. Specifically, we first regard the beginning of the playing Salsa music as the time of first beat $B(1)$, and then get the time of $B(2)$ by adding estimated SBI. For searching the next beats, we first get the candidate beat time $B(i-1)+EB$ through adding estimated SBI to the prior beat time. Then, we perform onset detection on the signal between the time $B(i-1)$ and $B(i-1)+EB+0.1$ to obtain the onset positions. Here, $0.1s$ is time of the tolerate delay for detected beats. From these onsets, we find the nearest onset to the candidate beat time. If the time difference between the nearest onset and candidate is less then $0.1s$ (i.e., time of the tolerate delay/advance), the nearest onset is regarded as the Salsa beat. Otherwise, the candidate should be identified as the beat. Additionally, considering the EB derived from the short period historical data may not be quite accurate, the value of EB would be continually updated to the mean value of the time intervals of detected beats after detecting 64 beats, which can make EB more accurate.

Algorithm 1: Online Beat Counting.

Data:
The received real-time audio signal at sampling rate $fs = 44.1kHz$: $X = [X(1), X(2), X(3)...]$;
The estimated value of Salsa Beat Interval: EB;
The length of window frame: $Win = (EB + 0.1) * fs$;
Result:
Time of Beats: $B = [B(1), B(2), B(3)..., B(i), ...]$;
Real-time Beat Counting Voice and Vibration Prompts;

begin
  $B(1) = 0s$. Insert Prompts at $B(1)$;
  $B(2) = B(1) + EB$. Insert Prompts at $B(2)$;
  if $i \geqslant 64$ **then**
    $EB = \frac{\sum_1^{i-1} B(i+1)-B(i)}{i-1}$;
    `// Replace` EB `with the mean value of previous SBIs`
  end
  while $i \geqslant 3$ **do**
    Perform Onset Detection on $X[B(i-1)*fs) : B(i-1)*fs) + Win)]$;
    Find the nearest onset position N around $B(i-1) + EB$;
    if $|N - B(i-1) - EB| \leqslant 0.08$ **then**
      $B(i) = N$. Insert Prompts at $B(i)$;
    else
      $B(i) = B(i-1) + EB$. Insert Prompts at $B(i)$;
    end
  end
  `// The units of` EB, B, N, `are all second(s);`
end

VI. Performance Evaluation

In this section, we evaluate the beat tracking accuracy of both *Offline Beat Counting* algorithm and *Online Beat Counting* algorithm in our system.

A. Experimental Methodology

We generate a test dataset which consists of 40 Salsa music audio files with the ground truth annotations of Salsa beat timings. Each file is an entire Salsa song which is around 5 minutes in length, stereo, and $44.1kHz$ sampling rate. Also, the files are annotated by the professional Salsa dancers, by recording beat counting voice (i.e., beats 1, 2, 3, 4, 5, 6, 7, 8) to the audio files. We manually note down the ground truth annotations according to the beat counting voice in every music file.

Every music file in our dataset is fed into *Offline Beat Counting* algorithm which can automatically record the timings of the detected beats. To test the *Online Beat Counting* algorithm, the Salsa music file is played by the external speaker. Our system first listens for 10-seconds of the playing music to compute the estimated SBI. After 10-seconds playing, the *Offline Beat Counting* algorithm would record the beat timings in real-time. In addition, we also compare the beat tracking accuracy of our algorithms with other three beat tracking approaches: the context-dependent beat tracking system (i.e., [6]), the two-fold dynamic programming beat tracker (i.e., [5]) and the beat tracker using dynamic programming (i.e., [4]).

B. Evaluation Metrics

For evaluating the beat tracking accuracy against manual annotations, we use eight evaluation metrics from the beat tracking evaluation toolbox [17] which are generally adopted in beat tracking evaluation. First, Accuracy-based Metrics which can assess the total number of correct detected beats, not considering the metrical level, include F-measure, CemgilAcc [3], PScore [18]. Second, Continuity-based Metrics [6] includes CMLc (Correct Metrical Level with continuity required), CMLt (Correct Metrical Level with no continuity required), AMLc (Allowed Metrical Level with continuity required), AMLt (Allowed Metrical Level with no continuity required), which can manifest whether the detected beats are at the metrical level of Salsa beats and the continuity of the correct detected beats. Moreover, to visualize the beat tracking performance, we utilize Beat Error Histogram [19] to show the distribution of the beat error probability.

1) Accuracy-based Metrics:

- **F-measure.** which counts the number of true positives TP (correct detection within a tolerance window of $\pm 70ms$, false negatives FN (missed detection) and false positives FP (extra detection) and defines the beat accuracy as $\frac{2*TP}{2*TP+FN+FP} * 100\%$.
- **CemgilAcc.** which utilizes a Gaussian error function with a standard deviation of $40ms$ to evaluate the time error between the detected beat and the closest annotation, so that the detected beats lying exactly on the annotated beat timings give the greatest score. The beat accuracy then is measured by summing up these scores.
- **PScore.** which measures beat tracking accuracy by summing the cross-correlation between impulse trains of the detected beats and the ground truth annotations, considering deviations within 20% of the annotated beat interval as correct.

2) Continuity-based Metrics:

- **CMLt.** which finds the correct detected beats at the metrical level of Salsa beats (i.e., 1, 2, 3, 4, 5, 6, 7, 8) and calculates ratio of the number of correct beats to the length of annotation sequence.
- **CMLc.** which finds the longest segment of continuously correct detected beats at the correct metrical level and calculates the ratio of the length of the longest segment to the length of annotation sequence.
- **AMLt.** which finds the correct detected beats at twice or half the correct metrical level and calculates ratio of the number of correct beats to the length of annotation sequence.
- **AMLc.** which finds the longest segment of continuously correct detected beats at twice or half the correct metrical level and calculates the ratio of the length of the longest segment to the length of annotation sequence.

3) Beat Error Histogram: We compute the beat error as the difference between each annotation and the nearest detected beat. The beat error of every detected beat is then divided by the annotation interval, which normalizes the beat errors between -0.5 and 0.5. In addition, if any beat errors are beyond $[-0.5 \quad 0.5]$, they can be confined back into the range $[-0.5 \quad 0.5]$ using modulo arithmetic. Finally, we utilize these beat error data to populate a histogram. The beat error Φ is defined as [19]

$$\Phi(i) = \begin{cases} \frac{min_j(|a-b_i|)}{a_{j+1}-a_j} & ,j=1 \\ & , \\ \frac{min_j(|a-b_i|)}{a_j-a_{j-1}} & ,j>1 \end{cases} \qquad (5)$$

where b and a refer to the sequence of detected beats and annotations, respectively. b_i denotes the timings of the i^{th} beat. a_j denotes the timings of the j^{th} annotation.

C. Results

Figure 9(a) and Figure 9(b) present the performance of our *Offline Beat Counting* and *Online Beat Counting* algorithms, Devies [6], Jang [5] and Ellis [4] under the Accuracy-based Metrics and Continuity-based Metrics, respectively. As shown in Figure 9(a), both *Offline Beat Counting* algorithm and *Online Beat Counting* algorithm outperform all the other beat tracking systems based on the performance indices of F-measure, CemilAcc and P-score. That is, our system have capacity to find more correct beats than other systems.

Moreover, other systems also seem to perform well where the F-measure scores of Devies [6] and Jang [5] are 69.03% and 83.24%, respectively. However, as shown in Figure 9(b), once the constraints over the metrical level of Salsa beats are

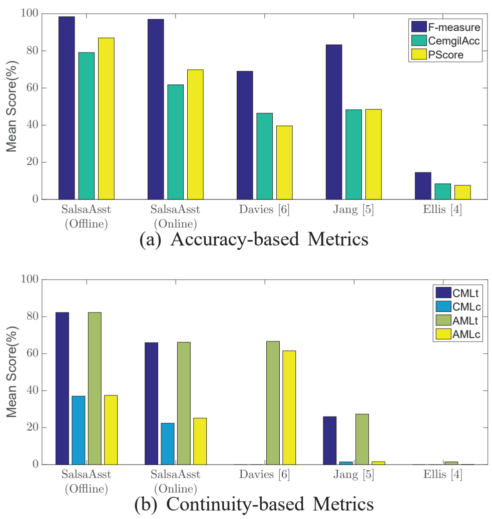


Fig. 9. Comparison of beat tracking algorithms under Accuracy-based Metrics and Continuity-based Metrics.

required, these systems are no longer accurate that the CMLt of Jang [5] is only 1.49%. The CMLt scores of Devies [6] and Ellis [4] are even equal to 0%, which indicates that the detected beats of these systems are all not at the metrical level of Salsa beats. Our proposed algorithms, however, are still accurate where CMLt of *Offline Beat Counting* algorithm is 82.28% and CMLt of *Online Beat Counting* algorithm is 65.96%, which demonstrates that most beats detected by our algorithms are exactly at the metrical level of Salsa beats, namely, our algorithms apply to Salsa music much better than others.

On the other hand, CMLc of our algorithms, reflecting the continuity of the correct detected beats at the Salsa beats metrical level, is only equal to 37.04% for *Offline Beat Counting* algorithm and 22.41% for *Online Beat Counting* algorithm. But the low CMLc merely indicates that the wrong detected beats of our system are scattered rather than centralized, which cannot prove the Salsa beat detection of our algorithms are inaccurate because our algorithms obtain a high score under CMLt. Additionally, we notice that the AMLt and AMLc of Devies [6] are both good and the AMLc of Devies [6] is even better than our algorithms. High AMLt and AMLc scores demonstrate that Devies [6] can accurately find the beats at the twice or half metrical level of Salsa Beats. Nevertheless, the CMLt and CMLc scores of Devies [6] are equal to 0%, that is to say, Devies [6] can merely detect the beats at the twice or half metrical level of Salsa Beats but is not able to find beats at the Salsa Beats metrical level.

Furthermore, as illustrated in Figure 10, we utilize Beat Error Histogram to visualize the performance of our algorithms and other three existing beat tracking approaches. The Beat Error Histogram is able to show the distribution of all beat errors in our dataset. For our purposed *Offline Beat Counting* algorithm, its Beat Error Histogram, as illustrated

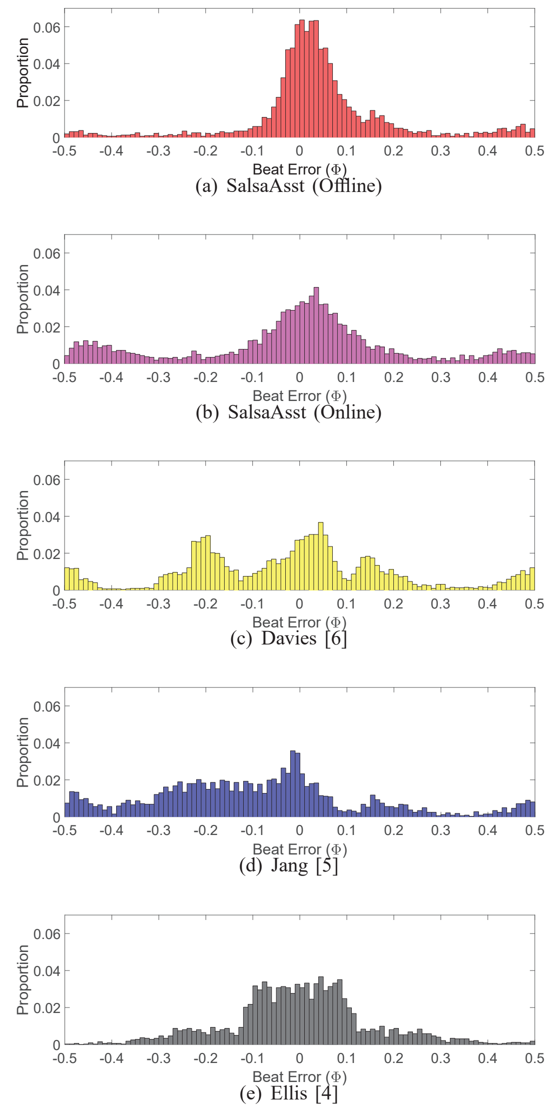


Fig. 10. Beat Error Histograms of our algorithms and other existing beat tracking approaches.

in Figure 10(a), merely exhibits a strong peak close to the error of 0, indicating most beats we detected are close to the ground truth annotations. For *Online Beat Counting* algorithm, as shown in Figure 10(b), although its Beat Error Histogram shows some errors concentrate on -0.4, most beat errors still focus around 0. Unlike our algorithms, the Beat Error Histogram of Devies [6] shown in Figure 10(c) has two strong peaks around ± 0.2 and other two strong peaks around ± 0.5, which indicates that a substantial part of detected beats are not matched with the ground truth. Likewise, Figure 10(d) and Figure 10(e) demonstrate that Jang [5] and Ellis [4] also perform badly due to the dispersive distributions of their beat errors.

Therefore, the results under 8 evaluation metrics clearly demonstrate the feasibility of our proposed system for accurately tracking Salsa beats from Salsa music signal comparing to other existing beat tracking systems.

VII. Discussion

In this section, we mainly discuss the potential improvement of the proposed *SalsaAsst* system, which will be presented in our future work.

A. Additional Assistance from Inertial Sensors of Mobile Devices

SalsaAsst has turned out to be a promising step towards enabling the mobile device to be a virtual assistant for dancing Salsa, which can help dancers to recognize the correct timings of Salsa beats. By exploiting more sensors such as inertial sensors (i.e., accelerometer and gyroscope) in mobile devices, it is possible to provide additional assistance to Salsa dancers. For instance, we can utilize the built-in inertial sensors of the smartphones that are worn with dancers to derive their dancing movements to see whether the movements are synchronized to the detected Salsa beats. It can be used to evaluate dancers' dancing synchronization during practice, and help them improve it by providing appropriate feedback if dancers have any error movements. In this circumstance, *SalsaAsst* will not only assist in identifying Salsa beats but also evaluate the actual dancing steps and correct their mistakes if any.

B. Remove the Requirement of Pre-listening in Online Beat Counting Mode

Different from other real-time beat tracking systems, the *Online Beat Counting Mode* of our system need to listen for a short time period (e.g., 10 seconds) of music before real-time Salsa beats counting, which is used for estimating Salsa Beat Interval (SBI) which determines the metrical level of Salsa music. However, such short period listening is just a small section of an entire Salsa music, which would not be inconvenient for dancers to use our system in practice. In the future, we will remove such a requirement of pre-listening by designing a quicker SBI estimation/update method which can quickly capture the SBI within just 2-3 music beats (i.e., less than 1 second) and update it accordingly.

VIII. Conclusion

This paper proposes a beat counting system, *SalsaAsst*, based on mobile devices (e.g., smartphones) to assist users to dance Salsa. *SalsaAsst* can perform rhythm deciphering and fine-grained beat tracking for Salsa music and generate beat counting voice or vibration prompts to remind the dancers of Salsa beat timings. Two alternative modes are developed to provide prompts with dancers in offline and real-time manners, respectively. Specifically, in the offline mode, *SalsaAsst* is able to process any Salsa music file to generate a new music file which possesses both original Salsa music and its corresponding beat counting voice prompts. In the real-time mode, our system leverages the embedded microphone on the mobile device to receive the acoustic signal of Salsa music and raise the real-time prompts (i.e., beat counting voice and vibration) along with the dance music. The proposed system can be used in many scenarios (e.g., home practice, dance studio and live performance), which can not only help Salsa beginners get accelerated learning process but also eliminate professional dancers' errors during their live performance. Extensive tests using 40 Salsa dance songs under 8 evaluation metrics show that our system can achieve much better Salsa beat tracking performance comparing to the existing beat tracking solutions.

IX. Acknowledgment

This work was supported in part by the National Science Foundation under grant numbers CNS-1514436 and CNS-1409767.

References

[1] D. Drobny *et al.*, "Saltate!: a sensor-based system to support dance beginners," in *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2009, pp. 3943–3948.
[2] G. Peeters, "Template-based estimation of time-varying tempo," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 158–158, 2007.
[3] A. T. Cemgil *et al.*, "On tempo tracking: Tempogram representation and kalman filtering," *Journal of New Music Research*, vol. 29, no. 4, pp. 259–273, 2000.
[4] D. P. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.
[5] F.-H. F. Wu *et al.*, "A two-fold dynamic programming approach to beat tracking for audio music with time-varying tempo." in *ISMIR*, 2011, pp. 191–196.
[6] M. E. Davies *et al.*, "Context-dependent beat tracking of musical audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007.
[7] A. Nakamura *et al.*, "Multimodal presentation method for a dance training system," in *CHI'05 extended abstracts on Human factors in computing systems*. ACM, 2005, pp. 1685–1688.
[8] M. Ohkita *et al.*, "Audio-visual beat tracking based on a state-space model for a music robot dancing with humans," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2015, pp. 5555–5560.
[9] T. Mizumoto *et al.*, "A robot listens to music and counts its beats aloud by separating music from counting voice," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1538–1543.
[10] S. Pietrobruno, "Embodying canadian multiculturalism: the case of salsa dancing in montreal," *Revista mexicana de estudios canadienses*, 2002.
[11] P. Renta, "Salsa dance: Latino/a history in motion." *Centro Journal*, vol. 16, no. 2, 2004.
[12] (2017) Bar (music). [Online]. Available: https://en.wikipedia.org/wiki/Bar_(music)
[13] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *The Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, 1998.
[14] S. Dixon, "Onset detection revisited," in *Proceedings of the 9th International Conference on Digital Audio Effects*, vol. 120. Citeseer, 2006, pp. 133–137.
[15] D. Deutsch, *Psychology of music*. Elsevier, 2013.
[16] C. E. Rice, "Human echo perception," *Science*, vol. 155, no. 3763, pp. 656–664, 1967.
[17] M. E. Davies *et al.*, "Evaluation methods for musical audio beat tracking algorithms," *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.
[18] M. F. McKinney *et al.*, "Evaluation of audio beat tracking and music tempo extraction algorithms," *Journal of New Music Research*, vol. 36, no. 1, pp. 1–16, 2007.
[19] M. E. Davies *et al.*, "On the use of entropy for beat tracking evaluation," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4. IEEE, 2007, pp. IV–1305.