NEW YORK UNIVERSITY

An Automatic
Music Performance Analysis System

Shashank Aswathanarayana

Submitted in partial fulfillment of the requirements for the

Master of Music in Music Technology

in the Department of Music and Performing Arts Professions in the

Steinhardt School of Culture, Education and Human Development,

New York University

Advisor: Dr. Morwaread Farbood

ABSTRACT

The key to learning a new art form is to pratice, perform and rectify your mistakes. The presence of an instructor is to help you take the right steps forward by identifying your weak areas and help you strengthen them. Automating such process carried out by humans has often been the focus of research in computer science. In conjunction with this approach, this Master's thesis proposes and builds a new software for singers to practice and get feedback in the absence of an instructor. The approach taken here is novel in that it is a semi-guided tutorial system wherein the singer gets to select a score from a large databse to practice and get feedback on. The scores are grouped in different categories that can guide a novice through a practice session. This grouping and the database of scores is constructed using Indian and Western vocal training methods, thus making it a really robust system.The feedback is provided in visuals and numbers based on pitch and tempo of the performance. Extracting such information from a sung voice is not a trivial problem as seen later in the document. Music Information Retrieval approaches are evaluated and an optimal solution is narrowed upon and implemented. Lastly a small subjective evaluation is carried out wherein people explored the various features of the software and rate the software against various criteria. Experimental results show that the software built was helpful and easy to use, and one that could be easily introduced into the market.

ACKNOWLEDGEMENTS

This thesis has been my baby step into the field of research. It could not have been possible without the love, affection, support and guidance of countless individuals around me. First and foremost, I would like to thank my parents for their unwavering support and encouragement. While my parents have been the pillars of support, my thesis advisor Dr. Mary Farbood has been the guiding hand not only in this project but throughout my graduate education at NYU. She has pushed me to my limits and helped me realize my potentials. I would like to thank her from the bottom of my heart. I owe a special shout out and thanks to my friend, music collaborator and classmate Ethan Hein, with whom I discovered a new musical side to myself. Countless hours recording, chatting, and spending time with him has contributed immensely to who I am today.

The would also like to take this opportunity to extend a big thank you to all the NYU Music Technology faculty – Dr. Agnieszka Roginska, Dr. Tae Hong Park, Dr. Juan Bello, Prof. Tom Beyer and Dr. Kenneth Peacock.

A special thank you to Nidhi Shetty for helping me with last minute work when technology failed and turning in the document on time was doubtful. Thank you, to all friends who have helped me with inputs, inspiration, a cup of coffee and your ears - John Turner, Alex Marse, Patti Kilroy, Amar Lal, Jeet Paul, Rachel Wardell, Uri Nieto, Areti Andreopoulou, Andrew Madden, Michael Musick, Finn Upham, Tsaiyi Wu and Zhiyuan Lin.

I would also like to thank my roommates Akhilesh Manjunath and Akshay Kumar, who have listened patiently to me ranting over my work.

I would like to thank my friends back home in India. Without your encouragement, I would never have taken this step – Nachiket Panse, Vinay Nagaraju, Nitye Sood, Ayushya Khanna, Gaurav Navale, Ashaya Mahabaleshwara, Vibhu Umesh, Shreyas Krishna, Ashwin Kastury, and Sridhar Mullapudi.

Lastly, I heartfelt thanks all my friends who remain unnamed. Though this page does not have enough room for all of you, your contributions are as important and will remain close to me.

CONTENTS

LIST OF FIGURES

1. Introduction

"Music, in performance, is a type of sculpture. The air in the performance is sculpted into something" – Frank Zappa, American Musician

A key to learning and mastering a new art form is to practice, perform and rectify your mistakes. The presence of an instructor in such a scenario is vital, as they help you take the right steps forward by identifying your weaknesses and helping you strengthen them. This helps in shaping you into a good artist. Automating such processes carried out by humans has been the focus of research in computer science for the last couple of decades.

While research in computer science has focused on a number of different areas in music, from producing to recording and learning; little has been done to address the needs of singers. There are a plethora of software tools in the music production domain, which address some of the key issues taken up here and perform the necessary tasks, but these tools (digital audio workstations) can be intimidating and not user friendly for novice singers with limited computer skills. And since the purpose of a digital audio workstation is different from the problem at hand, the format in which the output is presented to the user is also not helpful.

This Master's thesis, An Automatic Music Performance Analysis System, looks at building a model for novice singers to be able to train themselves in a semi-guided manner and master the art of singing without the help of an instructor. Rather than just presenting the user with a blank page, this system has a database of pre-composed melodic sequences that the user can make use of to guide them through

a practice session. As with a regular practice session, the database has short one to three note sequences that a singer would rehearse at the beginning of their session for breath control and practice of holding a note for a lengthy period of time, and has complex melodic sequences the singer can practice to touch on other aspects. The model is called semi-guided as the user is allowed the freedom to navigate through the tool in a non-linear fashion and can pick and choose the sequences they want to practice in any order of their choice.

The development of this model required extensive research into the methods of practice of a singer, visualization of the form of feedback to be provided, and various other software and interface design paradigms. One must constantly ask the questions, is this design easy for the user to navigate? Is the feedback being provided helpful and is it being presented in a form easily read and understood by the user? With these questions in mind, the thesis looked at addressing the following issues:

- Studying, understanding and presenting the limitations present in music technology and education in the area of singers, from a technological standpoint.
- How these limitations can be bridged with the development of a software tool to help novice singers practice in a fruitful manner in the absence of a tutor.
- What aspects of the singer's performance are to be dealt with in order to make this a useful application?
- How does the Automatic Music Performance analyzer address these issues and what are the results obtained?

- What could be done in future to make this application better and contribute greatly in the learning process of a singer?

This thesis is divided into five chapters. The introduction in chapter 1 is followed by a discussion in chapter 2, of the previous work done, which addresses relevant topics such as pitch detection, score following and tempo tracking. The process of extracting key information from recorded samples of a singer is discussed, and the merits and demerits of some of the approaches taken in the past are dealt with. The score following, pitch detection and tempo tracking algorithms are particularly scrutinized as they form the crux of this thesis. Following this, the methodology and approach taken by the author in developing this model is discussed in chapter 3. Chapter 4 is dedicated to the results and discussion and the thesis concludes in chapter 5 with a brief note on future work that can be done.

2. Literature Review

This section of the thesis primarily discusses the motivation behind the thesis and the gap it is looking to fill. It does so by examining the problem at hand, and evaluating some of the relevant work done. The section starts off with a discussion on what Music Performance Analysis is and why we need it. Following which, there is a note on the key aspects of music performance analysis and the areas on performance analysis this thesis is concentrating on. The last three subsections are dedicated to the prior work done on the technical side of things that directly or indirectly address the topic of performance analysis. The salient features and drawbacks of the approaches taken before are discussed.

2.1 What is Music Performance Analysis and why do we need it?

More than at any earlier period in musical history, the contemporary scene in serious music in dominated by the performer (Lipman, 1990; Repp, 1992). A musical performance is the act of expressing musical ideas through the medium of sound (Kohut, 1985). Music performance analysis, therefore, is the dissection of these musical ideas into their various mathematically, perceptually and cognitively relevant data and the study of the same. Analysis is an integral component in the improvement of a musical performance and it addresses the first of the two-part problem – diagnosis of a performance. This analysis can be divided into three groups: namely external, physical aspects; internal mental aspects and technicalities of the performance (Davidson & Coinbra, 2001).

Analysis of recorded performance in the past has focused on psychology and music theory (Cook, 1981). On the psychology side of things, the study of music performance has explored areas like emotion in music, and brain response to different musical performances and expressions. From the music theory standpoint, the focus has been on structural analysis of various musical pieces (Clarke, 1983, 1988). This thesis, on the other hand, looks at music performance analysis as a music information retrieval problem. The aim is to extract key features of a recorded melody and analyze them with respect to various parameters. Thus the focus of this research is on the third aspect of musical performance mentioned above – technicalities of the performance.

We know the phrase, “practice makes perfect”, but this is not necessary true in all situations. If one practices mechanically or cannot distinguish between the errors and corrects, then one cannot become perfect with practice (Kohut, 1985). This statement should be modified to “correct practice makes perfect”. For correct practice, as a novice, one needs guidance. He/she must constantly be made aware of the mistakes and guided in the right direction. Therein lies the need for an analysis system.

2.2 Key aspects of Music Performance Analysis

There are two basic aspects that define a music performance: a *normative* aspect that represents what is expected from a competent performer and is largely shared

by different artists, and an *individual* aspect that differentiates performers (Repp, 1992). The individual aspect is something to be worked on in isolation by the artist to set himself apart from the rest. The normative aspect, however, is narrower is scope and by and large the same for all artists. For singers, these two aspects can be expanded into the following factors, which are deterministic qualities during performance analysis. The factors are,

- Accuracy of notes sung.
- Range of the singer.
- Tempo consistency.
- Confidence in content. This includes clarity of voice and lyrics.
- Gestures, expressions and interaction with audience and fellow musicians.

The first three factors can come under the larger bracket of normative aspect while the last two come under the bracket of individual aspect of a performance. Automation in the tutorial aspect of performance can focus on the normative aspect since it is uniformly measurable across different performers.

Accuracy of notes sung and range of the singer boil down to the fundamental concept of pitch estimation, and tempo consistency is measured using tempo estimations. Sections 2.3, 2.4 and 2.5 are dedicated to pitch while tempo is discussed in section 2.6.

2.3 Pitch Detection

The goal of pitch detection is usually estimating the fundamental frequency (f_0), as pitch is a perceptual aspect of periodic and quasi-periodic sound objects (Park, 2010). Mathematically we can define a periodic signal as

$$x(t) = x(t + T_0), \forall t$$

Where T_0 is the fundamental period

$$\therefore Fundamental\ Frequency, f_0 = \frac{1}{T_0}$$

The task of fundamental frequency estimation becomes difficult with music due to a number of reasons. Some of these are:

- The spectral structure of music is quite complex in that, along with the fundamental frequency, there exist a number of different harmonics. These make the sounds richer and the sensation of pitch improves perceptually (Truax, 1978), however, it could add noise to the result during computation of fundamental frequency. Further complicating the problem is the issue of quasi-periodicities.
- At times, the fundamental is completely missing from the data. This is known as the missing fundamental phenomenon. In this case, due to the nature of sound, we are able to perceive the fundamental frequency as the pitch, but it proves to be a difficult problem computationally.
- A technicality that also proves to be a problem sometimes is the fact that pitch is a perceptual quantity while frequency is an absolute one.

- Transient, temporal variations along with ambiguous events also prove to be a problem during fundamental frequency estimation.
- Polyphonies in the form of overlap or harmonicity can often be an issue during fundamental frequency extraction.

Despite all the aforementioned issues, the advancement in digital signal processing has given us a number of different approaches and efficient algorithms to compute the fundamental frequency of a sound. These algorithms can broadly be classified into two groups namely

i. Frequency-domain approaches
ii. Time-domain approaches

2.3.1 Frequency-domain Approaches

Pitch detection using frequency-domain approaches primarily involves converting the time domain audio signal into the frequency domain using Fourier transform and then applying some estimating/tracking logic to extract f_0. Common frequency-domain approaches include Harmonic Product Spectrum, Cepstral Analysis and Spectrum Autocorrelation.

The harmonic product spectrum approach attempts to make use of the harmonic nature of music signals while estimating the fundamental frequency of the signal (Schroeder, 1968). The harmonic product spectrum takes the Fourier transform of a signal and down samples it of various factors. Then, when the different spectra and multiplied together, the fundamental frequency can be estimated by the presence of a strong peak in its location. The concept behind this is

as follows: As we know, the harmonics present in a signal are integral multiples of the fundamental frequency. Thus when a signal is down-sampled by an integral number, say 'n', then the 'nth' harmonic will align with the fundamental. Thus when this is multiplied with the original spectrum, the alignment will give rise to a peak at the location of the fundamental frequency. While this algorithm is simple in approach and implementation, it has the critical shortcoming that it does not handle quasi-periodicities well.

In Cepstral Analysis, the Fourier transformed version of the signal is treated as though it is the signal itself and analysis is carried out. To be more precise, the log magnitude spectrum of the signal is treated as the signal and the (I)DFT of this signal is taken to obtain the fundamental frequency. This assumption and treatment of frequency domain in signals and naming convention is explained by its developers, Bogart, Healy, Tukey (1963), who say, "..we find ourselves operating on the frequency side in ways customary on the time side and vice versa." This method worked well for echoes and speech signals but was not very good at estimating the fundamental frequency for music.

Lahat, Niederjohn and Krubsack (1987) came up with a modified autocorrelation based algorithm called spectral autocorrelation for pitch estimation. This algorithm applies the autocorrelation function on the magnitude spectrum of the signal. This method was particularly useful, as any frequency components with spectral interval between them have a corresponding f_0 associated with it. This makes the spectrum shift invariant, which is very useful in handling quasi-

periodicities. However, the spectral interval, which makes this system immune to quasi-periodicities, also provides a major drawback for it. This drawback comes in the form of f_0 doubling. The magnitude spectrum is periodic at twice the fundamental frequency and hence gives rise to f_0 doubling errors.

2.3.2 Time-Domain Approaches

One of the earliest and simplest pitch detection algorithms was zero-crossing rate. This method involved computing the number of times the signal crosses zero per unit time (Kedeem, 1986). The simplicity of this method is reason enough to believe that this is not an efficient approach for pitch estimation. This method fails for spectrally complex waveforms like music, as these rarely have only one event per cycle. Thus one can find multiple crosses in a cycle. This naturally leads to spurious results.

Other commonly used time domain approaches are based on the autocorrelation function. The autocorrelation function is a measure of the cross-product similarity of the signal across time. Brown and Zhang (1991) found these models to be the most frequently used for fundamental frequency estimation as they give reliable and accurate results.

With autocorrelation, the similarity of the function is measured with delayed versions of the signal. Mathematically,

$$r_{xx}(l) = \sum_{n=0}^{N-1} x(n) * x(n+l)$$

$$where\ l \rightarrow lag\ in\ samples$$
$$n \rightarrow sample\ index$$

$$N \rightarrow Length\ of\ analysis\ frame$$
$$r_{xx}(l) \rightarrow Autocorrelation\ value$$

Therefore, maximum correlation is obtained at no delay position (l = 0) when the signals are identical. This property will repeat at multiples of the period of the signal, hence we get peaks at T_0, T_1, T_2, T_3 etc. This in turn gives us the fundamental frequency of the signal.

The main drawback with this method is the case of quasi-periodic signals. The peaks do not occur at exact multiples of the period of the signal, hence leading to slight errors in the estimation of the fundamental frequency. Since, most music signals are quasi-periodic in nature, this drawback gets amplified in the case of music.

The solution for this drawback is seen in the algorithm developed by de Cheveigné and Kawahara (2002). Their algorithm, called YIN, is based on the autocorrelation method described above, with tweaks and modifications in the normalization of the autocorrelation function. De Cheveigné and Kawahara (2001) showed the YIN algorithm is 10% more efficient than the next best algorithm in a comparative study that included 9 other state of the art algorithms (which used both time domain and frequency domain approaches) and spanned 4 different speech and music databases. The YIN algorithm is therefore used for fundamental frequency extraction in this thesis and is explained in greater detail in the following chapter.

2.4 Issues with the Sung Voice: Need for Score Following

In the previous section, a number of methods for estimating the pitch of an audio signal were discussed. This section focuses more on the issue at hand: estimating the pitch of a sung voice. This problem is not as straightforward as directly applying one of the algorithms discussed above.

Musical audio signals can be classified into four groups (Bello et al. 2005): pitched percussive sounds (e.g. piano), non-pitched percussive sounds (e.g. drums), pitched non-percussive sounds (e.g. violin) and complex mixture (e.g. pop recording). Percussive sounds, pitched and non-pitched, in general have hard onsets and high-energy differences across the signal, which makes it easy to process and analyze them using different algorithms. Non-percussive sounds on the other hand have soft onsets and low-energy differences across the signal, which makes the process of analysis more complicated. Complex mixtures have a combination of percussive and non-percussive sounds and fall under the category of polyphonic and multiple pitch estimation, which is beyond the scope of this thesis.

The sung voice comes under the group of pitched non-percussive sounds. It therefore has the characteristics of soft onsets and low-energy differences. Apart from these, the sung voice also has the following characteristics, which make the task of pitch estimation a non-trivial problem.

- Extended transients – In singing, the attack of a note can be long, especially in a slow tempo as the singer attempts to build up the note. In other cases, the sustain might be long. Both these cases result in extended transients

leading to the difficulty in locating the exact location of onset of note and thereby causing computation problems in pitch estimation.

- Vibrato – The fluctuation of a pitch to add expression to a music performance is called vibrato (Sundberg, 1987). This effect causes difficulty as one has to try and separate the intentional pitch drifts from the unintentional ones, which is computationally very difficult as the musical knowledge and judgment of a person is lacking during retrieval.
- Portamento – The effect used by a singer to glide from one note to the other (either ascending or descending) is called portamento. This effect is also known as glissando and is sometimes interchangeably used with portamento. However, the distinction is that portamento is used when the phenomenon is more continuous as in the case of singing, while glissando is used when the effect is discrete as in the case of a piano. Gliding from one note to the other will affect the pitch estimation in that; it involves singing of intermediate frequencies and some that are between two semitones, which cannot be classified. This problem also leads to the case of extended transients discussed above.
- Spectral tilt variation – This is another attribute of singing that varies with loudness (Macon, Link, Oliverio, Clements & George, 1997). The source spectrum varies (downward tilt) with the crescendo of the voice (Bennett and Rodet, 1989). Macon et al. (1997) also found that breath fluctuation resulted in the modification of the spectrum and manifested itself as high frequency noise. This is very crucial as noise in the frequency of the signal will affect the estimation of the pitch.

All these factors bring down the accuracy of the pitch estimators discussed above. In order to increase the efficiency of the pitch estimation, one would need to use the f_0 estimators along with score followers: what they are, how they can be useful in improving pitch estimation of sung voice, what are the different kinds of score followers, and their advantages and disadvantages are all answered below.

2.5 Score Following

The process of either providing online or offline alignment of a pair of audio data is called score following (Cont, Schwarz, Schnell & Raphael, 2007; Schwarz, Orio & Schnell, 2004; Orio & Déchelle, 2001; Puckette, 1995). The pair of audio data can be of various kinds. Depending on the type of pair, the score follower is given the corresponding name. For instance, if the pair are both audio signals, then the task achieved is audio-audio alignment. If it is one audio signal and one MIDI signal, then the alignment is MIDI-audio alignment and so on. Alignment of audio data, i.e. score following, lends itself to a number of different musical applications, for example: computer aided musical accompaniment, musical performance analysis – compare expressive performances of the same musical piece by different performers.

Score following/alignment of two sequences has been a research topic for several decades. Research in the alignment of sequences started with speech recognition and molecular genetics in the late 70s and early 80s. This topic was extended to alignment and synchronization of musical sequences in 1984, when the first two papers appeared (Orio & Déchelle, 2001). Early approaches involved some sort of string matching techniques with heuristics to prevent errors in decision

making at real-time. Puckette (1990) adopted a unique approach to compare incoming events with a set of expected events and choose the first exact match as the event for alignment. In another approach, two pitch trackers were used, one fast (imprecise) and one slow (reliable) to account for imprecisions in note detection (Puckette, 1995). Since these early approaches, the topic has gained widespread attention and progress has been rapid. The most popular approaches taken by researchers to achieve score following are:

- Dynamic Time Warping or DTW (Orio & Schwarz, 2001; Dixon, 2001, 2005; Devaney, 2009).
- Hidden Markov Models or HMMs (Raphael, 1999; Orio & Déchelle, 2001; Schwarz, Orio & Schnell, 2004; Cont, Schwarz & Schnell, 2005; Cont, 2006, 2010).
- Chroma onset feature extraction (Hu, Dannenberg & Tzanetakis, 2003; Dannenberg & Hu, 2003; Müller, Kurth & Clausen, 2005; Ewert, Müller & Grosche, 2009)

2.5.1 Chroma onset feature extraction

This method for achieving score following came into prominence in early 2000s. Initially the audio vector is processed to obtain the chromagram – a sequence of chroma vectors. This is the representation of the audio's pitch class profile. It gives the distribution of the signal's energy across the pitch class set. The next step is to obtain the chromagram for the audio/MIDI file with which the incoming audio data is to be synchronized. Finally, audio matching/synchronization is achieved by calculating the Euclidean distance between the sequences of chroma vectors. Euclidean distance gives information of how closely the sequences match, with a

distance of 0 indicating perfect match while a large value indicating little or no correlation between the vectors. The matrix used to store the Euclidean distance values is called the similarity matrix. Traversing this matrix by following the least value path eventually determines the score matching/following. This process is explained further with the help of the following diagram.

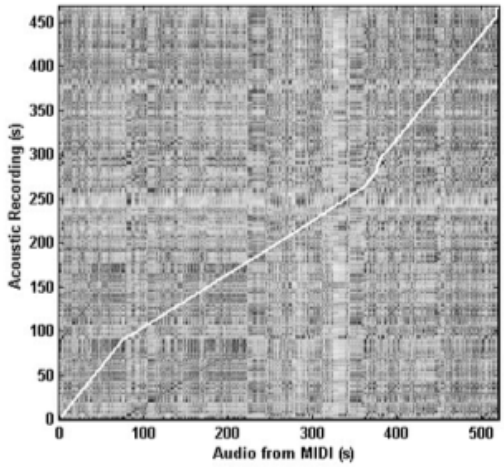


Figure 2.1 Similarity matrix between an Audio recording and its MIDI representation

The figure above shows a similarity matrix (final stage of score following after chroma feature extraction) of an audio recording and its MIDI version. The MIDI version in this test was artificially varied in tempo such that the acoustic recording and MIDI versions do not have the same tempo. The chroma vectors of the acoustic recording are along the y-axis while the chroma vectors from the MIDI file are along the x-axis. From the figure, we observe that the traversed path (white line) is approzimately along the diagonal, but not exactly. From this, we can infer that the two files are very similar, pretty well matched, but are not the same. This difference is owed to the difference in tempi between the files.

The advantage with chroma vectors is that they depend on pitch classes of strong partials (Dannenberg & Hu, 2003). Additionally, chroma based features are

invariant to dynamic variations (Ewert et al., 2009). The disadvantage of this method is that timing information is lost. Also, more critically, if different pitch classes align, the one that is stronger will take prominence and key information may be lost. This can be particularly harmful if the eventual goal is pitch detection. One also needs to preserve the timing information for pitch detection and thus this method falls short.

2.5.2 Hidden Markov Models (HMMs)

The Hidden Markov Model is a statistical based model in which the system is divided into various states. In score following, Orio & Déchelle (2001) defined a very robust two-level model, one that models the performance (higher level) and one that models the signal (lower level).

At the higher level, events related to the performance, i.e. notes, rests, chords etc. are modeled into normal states, n-states (events correctly played), and ghost states, g-states (events with mismatch between score and performance). At the lower level, the incoming signal is modeled with states related to event features like attack, sustain and silence at the end. With these states, a self-transition probability matrix is drawn. At the decoding stage, a modification of the classic Viterbi Algorithm is used. The principle, however, remains the same – find the lowest computational cost path from the self-transition matrix. The author does not present the topic in further detail, as this approach is not eventually used in this thesis. For curious readers, the author would like to direct to Orio & Déchelle (2001) and Schwarz et al. (2004).

The HMM approach has a lot of advantages. It is very robust and can handle polyphony in the music very well (Raphael, 1999; Schwarz et al., 2004; Cont, 2006). Other musical features like vibrato and trill can also be modeled using HMMs. This would seem like the ideal approach to use for the sung voice. However, this model is complex and therefore computationally expensive. Additionally, it presents overfitting problems. This thesis deals with monophonic score following, so such general and computational models are not necessary. As seen later, the DTW approach used is conceptually related to the HMM approach, and thus one does not lose out on accuracy or robustness by discarding this method.

2.5.3 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) is a well-known technique for the alignment of two sequences. One of the first methods to provide alignment (Müller, 2007), this algorithm came to prominence in 1970s when it was used to compare different speech patterns in speech recognition systems (Dixon, 2005; Müller, 2007). Since Dannenberg (1984), DTW has found its application in the music information retrieval realm for tackling various music problems, from simple audio-audio alignment to more complex, automatic accompaniment in computer music.

The main objective of the DTW algorithm is to find an optimal alignment between two time dependent sequences. These sequences are warped in a non-linear fashion to obtain alignment. Mathematically speaking, if the two sequences are $X := (x_1, x_2, x_3 \ldots, x_n)$ of length N and $Y := (y_1, y_2, y_3 \ldots, y_m)$ of length M and we have a feature space F such that $x_n, y_m \in F$ for $n \in [1{:}N]$ and $m \in [1{:}M]$. To compare X and Y in the feature space F, we have some sort of a cost measure 'c' such that if x and y

are similar, the cost $c(x,y)$ is small and if $c(x,y)$ is large, x and y are dissimilar. This evaluation of the pairwise cost measure of the elements of X and Y leads to a cost matrix (similar to the one discussed in section 2.5.1). This cost matrix 'C' is defined as, $C(n,m) := c(xn,ym)$. The goal for optimal alignment then becomes to find the minimal cost path. An example of how warping to sequences to obtain alignment is shown is the figure below (Taken from Müller, 2007). Further elaboration of this algorithm is provided in chapter 3 (Design and Implementation) as the author has chosen to implement this algorithm in this project.

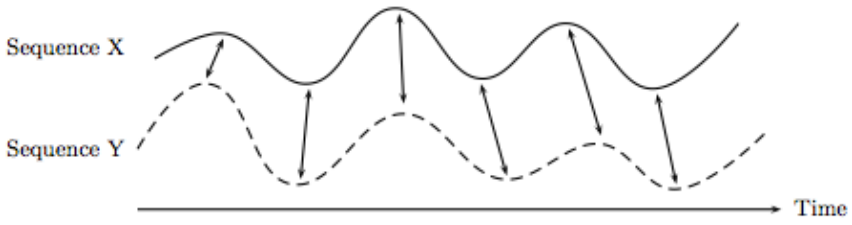


Figure 2.2: Time alignment of two time-dependent sequences. Aligned points are depicted using arrows.

2.5.4 Usefulness of score following in pitch estimation for a sung voice

In the previous sections, unique characteristics of the sung voice are discussed and the issues that come along with these characteristics with respect to pitch estimation are also presented. This section deals with how a good score following algorithm when implemented along with a pitch estimator helps improve the accuracy of pitch estimation of the sung voice.

Acoustical features like vibrato and portamento can be detrimental and hamper the pitch estimation process. Time alignment using a score following algorithm helps

identifying the location of the onset of a specific location, which narrows down the window for error in pitch estimation due to incorrect event onset detection. Further, score following enables us to model acoustical properties like breath, transient, sustain/steady state. Modeling transients helps determine the correct identification of the location of the voiced section and where the note begins (which in turn helps in estimating pitch at that location). Sustain/steady state information gives us the duration of the note and therefore the time window to apply the pitch estimation on. Thus better classification of the sung voice at the note level is achieved with score following, which in turn leads to reduction in errors during pitch estimation.

2.6 Tempo Tracking

Tempo tracking is the process of determining the global tempo or speed of a piece of music (McKinney, Moelants, Davies & Klapuri, 2007). Research in the extraction of tempo related information from music began in the 1970s. Longuet-Higgins and Steedman (1971) extracted the meter and tempo from a score representation of notes. This was followed by numerous attempts in automatic pulse detection in music. Pulse detection involved beat level analysis and hence tempo tracking is often confused with beat tracking. Tempo tracking should be distinguished from beat tracking although the terms are similar and are used interchangeably at times. Tempo tracking is a more global process while beat tracking is locating an individual beat. Extraction of tempo information from music can be done without having knowledge of the position of individual beats and thus the two tasks differ. Summaries of some state-of-the-art tempo tracking algorithms are discussed to find an optimal solution of the problem at hand.

McKinney et al. (2007) in a study evaluated a number of tempo and beat tracking algorithms. Their evaluation consisted of running the algorithms with a varied dataset, from western classical music to jazz to pop. They also tested the algorithms in percussive music and non-percussive music conditions. This means, music with percussive sounds present/absent in it. Music with different meters was also tested. Lastly, the algorithms were tested with music of different tempi. This shows that the evaluation was thorough and the results have significant weight.

In the tempo extraction part of the evaluation, McKinney et al. (1971) found that the algorithm implemented by Klapuri (Klapuri, Eronen & Astola, 2006) had the best P-score and statistically outperformed all algorithms barring Davies' (Davies & Plumbley, 2007). Davies' algorithm on the other hand outperformed all other algorithms except Alonso's (Alonso, Richard & David, 2007). The mean p-scores across musicological factors shows that Klapuri (2006) and Davies & Plumbley (2007) are very close to each other at 0.8 and 0.78 respectively. Davies' algorithm statistically outperformed Klapuri's. Therefore, logically these two algorithms should be closely studied in order to make an informed choice between the two.

2.6.1 Tempo tracking (Klapuri et al., 2006)

This paper deals with the meter analysis of music at three levels – tactum level, which is a temporally atomic pulse level; tactus level, which corresponds to the tempo of the piece; and the musical measure level. This makes the approach both wholesome and robust. This three level method proposed by Klapuri et al. (2006) is shown in the figure below.

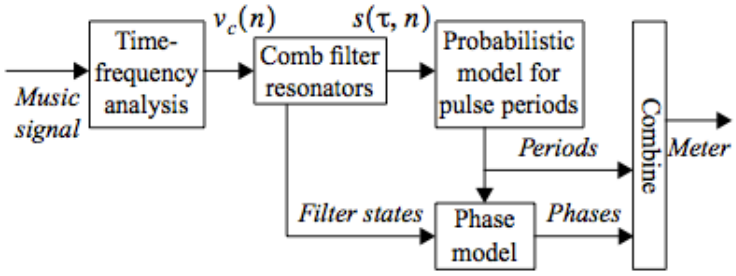


Figure 2.3: Method for tempo extraction proposed by Klapuri et al. (2006)

The time-frequency analysis part gives information about the musical accents present in the signal. Feature extraction from the time-frequency analyzed signal is carried out to obtain the pulse periods. This is done using comb filter resonators, as shown above. The pulse periods are then run through a probabilistic model. This helps join the frames of tactum, tactus and measure level analysis. Joining the frames is thereby used to combine and estimate the temporal continuity of the system and model it. These models help reduce errors and obtain more stable tempo tracking. The primary reason for that is the probabilistic models used are built using prior musical knowledge, which helps eliminating some computational errors.

2.6.2 Tempo tracking (Davies & Plumbley, 2007)

The tempo tracking algorithm developed by Davies and Plumbley has two main stages (onset detection and autocorrelation) in extracting the global tempo information. They proposed that the global tempo could be determined by applying a global autocorrelation function calculated across the onset detection function. This process is captured by the block diagram give below.

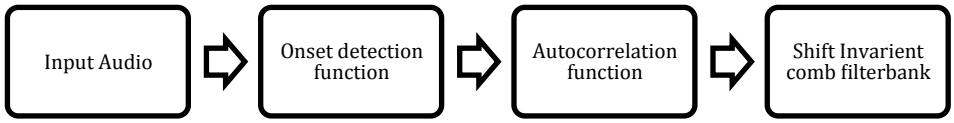


Figure 2.4: Block diagram of tempo tracking algorithm by Davies and Plumbley

The complex domain onset detection method (Bello et al., 2004) is chosen as the method of detecting the onsets of the music. The local mean is then removed and half wave rectified to keep the function positive. The autocorrelation operation is then performed on the onset detection function. This emphasizes the fundamental frequency and its harmonics. Then in order to compute the periodicity, the autocorrelation function is then dot multiplied with a weighted comb filterbank. This not only brings out the periodicity of the music but also makes the function shift invariant. The comb filterbank used here is weighted such that the mid-tempi are emphasized. This is done to restrict the tempo to be within the 80-160bpm range and to not have many outliers during tempo estimation. During tempo computation one averages over the entire file and outliers can throw the result off the actual value. Secondly, most music has its tempo within that range and hence emphasizing it gives the algorithm a better chance to recover the correct value.

As stated previously, these algorithms described outperformed other state-of-the-art algorithms as tested by McKinney et al. (2007). A drawback of both these methods is that the comb filters used are perceptually weighted in the choice of tempi. This restricts the tempi to the mid-tempi range. While this prevents outliers, it also causes major problems in tempo estimation for slow and fast tempo music, leading to errors at the metrical level. Grosche and Müller (2009) came up with a

new method for tempo tracking that outperformed the aforementioned algorithms. This algorithm hereby referred to as the Grosche algorithm in the rest of the document is used for computation of tempo in the thesis. It is therefore explained in greater depth in Chapter 3 (Design and Implementation).

3. Design and Implementation

The development of the Automatic Music Performance Analyzer consists of the following parts.

- Implementation of a good pitch-tracking algorithm. The YIN algorithm was selected for this purpose.
- Implementation of a good score following algorithm, which good be combined along with the pitch-tracking algorithm to enhance the efficiency of the pitch tracker as the sung voice causes issues discussed earlier. The DTW algorithm was selected in this case.
- Implementation of a tempo tracking algorithm. For this software, the Grosche algorithm was selected for tempo tracking.
- Creation of a database of scores for the singer to practice from. This had to be carefully designed to be able to cater to a novice (the primary target audience) as well as an experienced singer.
- An effective way to record and store data from a singer.
- An easy tool to access the performances for playback and evaluation purposes.

Implementation of the three algorithms mentioned above was done in matlab. The database was created using the online music notation software, Noteflight, which is a powerful tool to create, edit, playback and download scores in multiple formats. The Max/MSP software was used for recording, accessing and playing back the recordings and as the front end UI. The development of this software was

restricted to the prototype stage only in order to be able to get feedback from novice and experienced singers as to the drawbacks of the software, so that they could be addressed in its final development. The ultimate, long term goal of this project is to make a web-based application running entirely in the browser using web-audio standards like HTML 5. This would give the users the opportunity to create their own accounts as well as add to the community by inserting their scores and compositions resulting in a larger database.

3.1 Pitch Tracker

As mentioned earlier, the YIN algorithm is used for pitch tracking. This algorithm is very robust and outperformed other pitch tracking algorithms across a varied dataset (De Cheveigné and Kawahara, 2001).

Developed by De Cheveigné and Kawahara (2002), the YIN algorithm is a modification of the Autocorrelation algorithm discussed earlier. It addresses the drawbacks of the ACF function in that the octave errors occurring at large T_0 values (where T_0 is the period of the signal) are corrected using the difference function.

3.1.1 Architecture

The key steps in building any pitch tracker can be broken into the following block diagram. It involves the same steps and only the detection function in each algorithm varies.

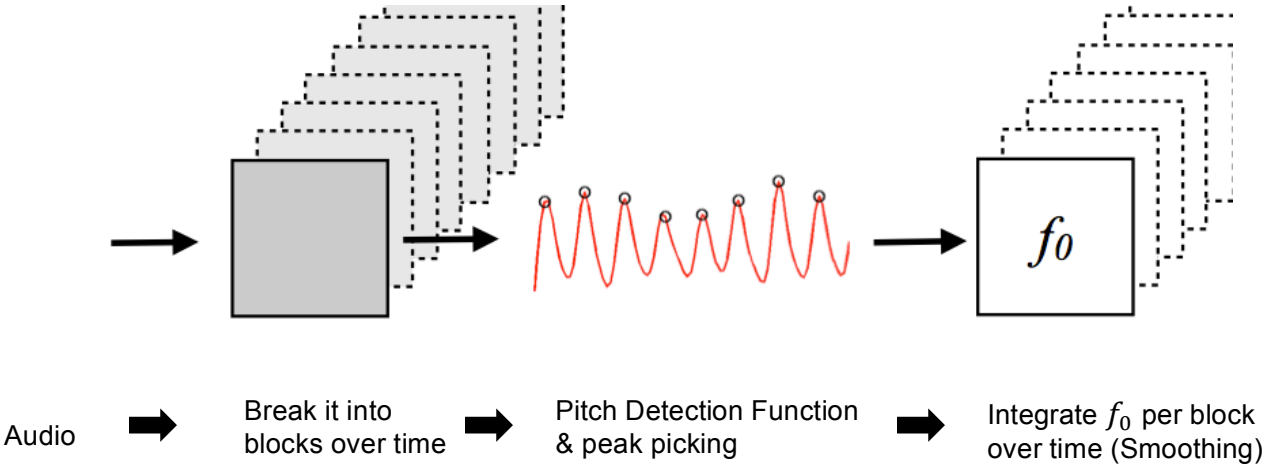


Figure 3.1: Flow diagram of the steps in pitch tracking

As can be inferred from the block diagram, the pitch detection function used will determine how good the tracker is. The incoming audio signal is broken up into blocks over time (windowing the signal). This is carried out to work on shorter chunks of audio and smoothen it to remove outliers. Second stage is the crux of the algorithm, when the pitch detection function is applied to the audio signal such that the peaks would give information about the fundamental frequency. Performing peak picking at this stage gives the estimated fundamental frequency of the block of audio. In the final stage, the fundamental frequency estimates of each block are integrated over time to obtain the final result. This process is also called smoothing. The mathematics of the YIN algorithm used as the pitch detection function is discussed below.

3.1.2 Mathematics of the algorithm

For f_0 computation, the squared difference function is calculated using the formula,

$$d(l) = \sum_{n=0}^{N-l-1} (x(n) - x(n+l))^2$$

$$where\ d(l)\ \rightarrow Squared\ difference\ function$$
$$N\ \rightarrow Length\ of\ analysis\ frame$$
$$l\ \rightarrow Lag\ in\ samples$$

This gives us the energy of the signal. However, the equation fails when the signal is periodic or quasi-period with period Examination of this equation informs that for signals with period l, when the difference function becomes 0 at $l = 0\ (l_0)$ and multiples of (l_0). This gives rise to a zero bias, which can affect the performance of the algorithm. To account for this, a normalizing factor is introduced such that,

$$\hat{d}(l) = \begin{cases} 1, & if\ l = 0 \\ d(l) \Big/ [(1/l) \sum_{j=1}^{l} d(j)], & otherwise \end{cases}$$

This normalization not only accounts for zero lag bias, but also helps in increasing the frequency limit of the search range (De Cheveigné and Kawahara, 2002). The $\hat{d}(l)$ function remains large for low lag values and drops below 1 when the difference falls below average. The next step is to find the peaks or in this case troughs as the difference function falls and the minimum values are aligned to the fundamental frequency. Trough peaking yields the point when its neighbors are larger than it. The index of this value indicates the sample at which the function has a minimum. With knowledge of the sampling rate (f_s) of the audio signal, the fundamental frequency can be estimated as

$$f_0 = {f_s}/{lag\_index\_of\_trough}$$

3.2 Score Follower

Synchronization can be of multiple kinds. Generically speaking it's the determination of the position in one representation given the corresponding representation in another representation. In this case, MIDI-audio alignment is dealt with. Whenever

alignment is attempted, algorithm design has trade-off between robustness and temporal accuracy. With these in mind, and the discussion in section 2.5, the DTW algorithm is chosen for implementation. Although the HMM approach is much more advanced, is neglected as training of dataset is not essential in this case. The computational complexity of HMMs is not necessary. Also, Dannenberg and Ning (2003) pointed out that DTW is a particular form of HMM where cells in the matrix correspond to states, and the chroma vector distance serves as the output probability for a given state (emission probability). Thus, in off-line tracking using DTW, one is actually utilizing/implementing the robust HMM features, albeit in a slightly different manner, and thereby achieving the similar results without increasing the complexity of the system or making it computationally heavy.

As mentioned in section 2.5.3, the main objective of the DTW algorithm is to find an optimal alignment between two time dependent sequences. These sequences are warped in a non-linear fashion to obtain alignment. The steps involved in this process can be summarized using the following block diagram.

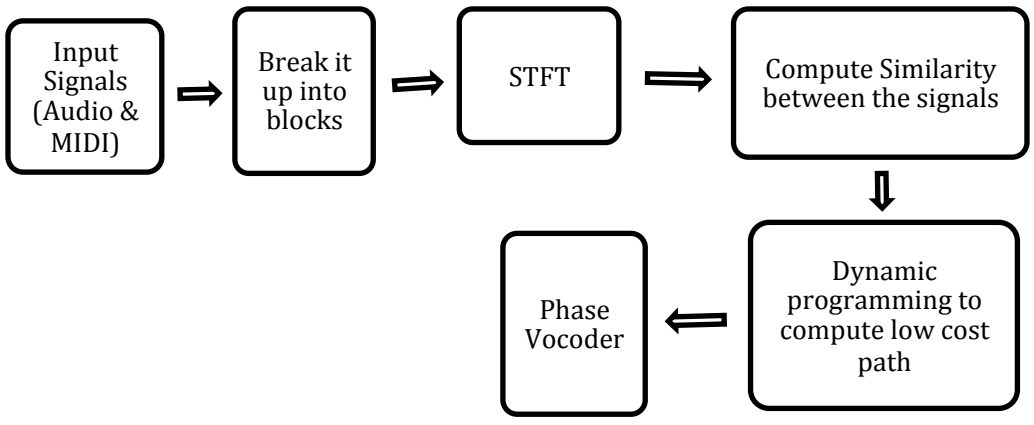


Figure 3.2: Flow diagram of DTW Score Follower

Initially, the signals are converted into the frequency domain for analysis. This is done by taking their STFTs (Short Term Fourier Transforms). The similarity matrix between them is then constructed using the cosine distance as the similarity measure. Dynamic programming is used to compute the lowest cost path. This is an iterative process where, for every point, the cost of its neighbors is computed and the least cost path is chosen. This can be visualized as shown in the figure below (figure borrowed from Dixon, 2005). The darkened gray squares represent the optimal path. As seen in the figure, for a point, say one, its neighbors, 2, 3, 4 are all computed before fixing on the lowest cost. In the implementation of this project, two neighbors are considered, $(i+1, j) and\ (i, j+1)$. Under ideal conditions, if the two input signals perfectly align with each other, the path will is the straight diagonal, but due to expressive nature of performances, the least cost path strays from the diagonal.

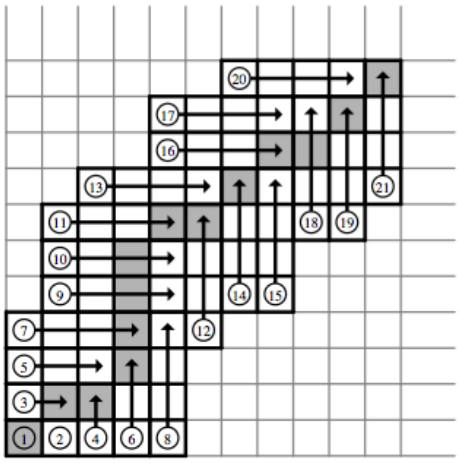


Figure 3.3: Optimal path construction using DTW algorithm

Following identification of the least cost path, one signal has to be time-warped in order to achieve alignment. The phase vocoder is used for this step, as the signal being dealt with is human voice. Once alignment is achieved, the exact location of the onset of a note is obtained, which helps in more efficient pitch tracking.

3.3 Robust Pitch Tracker using Score Following

As stated repeatedly, score alignment information has a positive impact on the pitch information extracted. It gives us exact time information about the onset of a note and the duration of silences, vibrato and other acoustical features.

The pitch of the performance is calculated using the weighted means approach. In this, the rate of change of fundamental frequency is computed and the means of the frames of analyses are weighted accordingly while matching the f_0 values computed for the modulated signal (DTW modified) and the unmodulated signal (Gockel, 2001). The weighting is done such that frames wherein the rate of change of f_0 is small are assigned a higher weight while frames with high rate of change of f_0 have a lower weight. The distinction between high and low rate of change of f_0 is set to 1.41 octaves/second based on the vibrato rate (Prame, 1994, 1997). Finally, the vibrato rate is calculated using the dominant frequency of the FFT of the pitch contour (Prame, 1994). These modifications to the f_0 computed using the YIN algorithm, yields greater pitch estimation accuracy (Devaney, Maandel, Fujinaga, 2012).

3.4 Tempo Tracking

The Grosche algorithm, used for tempo tracking in this software, is described below. Grosche and Müller (2009) say that extracting tempo information for signals with soft onsets is a challenging task. This is akin to the problem with extracting pitch information. However, unlike the pitch problem, a single solution for extracting the global tempo information from music has been developed as opposed to a combination of different algorithms.

3.4.1 Architecture

Grosche and Müller (2009) introduced a novel approach to determining the tempo of a music piece. Firstly, they derived a tempogram, which is obtained by performing local spectral analysis on a representation of the onsets of the signal. This tempogram gives an accurate representation of the local periodic information in BPM (beats per minute). The periodic information is then aggregated using local sinusoidal kernels to obtain the predominant local pulse of the signal. This gives a good representation of the local tempo of the signal. This process can be represented in the block diagram shown below.

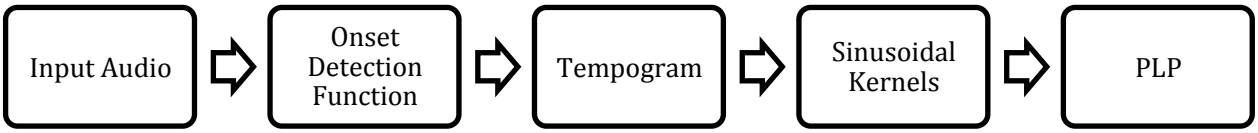


Figure 3.4: Flow Diagram of Tempo tracking

3.4.2 Mathematics of the algorithm

The first step in tempo estimation is the onset detection function. The spectral flux/ or change in spectral content is used to detect novelty. This is defined as

$$SF(n) = \frac{2}{N}\sum_{k=0}^{\frac{N}{2}} H(\,|X_k(n)| - |X_k(n-1)|\,)$$

$$H(x) = {}^{(x+|x|)}/_{2}$$

$$where, x \rightarrow input\ audio\ signal$$
$$X \rightarrow FFT(x)$$
$$H(x) \rightarrow Half\ wave\ rectified\ signal$$
$$N \rightarrow Length\ of\ audio\ frame$$
$$SF \rightarrow Spectral\ Flux$$

The signal is half wave rectified in order to only take the energy increases in the signal (Duxbury, 2002). Following this step is to perform local spectral analysis on this signal to obtain the tempogram ($\mathcal{T}$) with predominant local tempo information. For this, the DFT of the spectral flux function is taken. This DFT is range specific in order to limit the range of the tempo. Since the database had scores up to 180 BPM, and integral errors of the tempo must be accounted for, the tempo range is set from 30-400BPM. This means the DFT is taken for the range $\omega \in [30{:}\,400]/60 * fs_{SF}$ where ω is the frequency and fs_{SF} is the sampling rate of the novelty function. Choosing the frequency that maximizes the magnitude spectrum of each frame of the DFT of the novelty function (peak picking) gives us the tempo of each frame of audio, also called the local tempo.

The next step is the generation of the sinusoidal kernels, for which the phase of the kernel for each audio frame is computed by

$$\varphi_t = \frac{1}{2\pi}\arccos\left(\frac{Re\big(\mathcal{T}(t)\big)}{|\mathcal{T}(t)|}\right)$$

$$where, \varphi_t \rightarrow Phase\ of\ the\ frame\ 't'$$

$$\mathcal{T}(t) \rightarrow Tempogram\ of\ frame\ 't'$$

Using this phase information, an optimal sinusoidal kernel is constructed as,

$$\kappa_t(n) = W(n-t) * \cos(2\pi(\tau_t/60 * n - \varphi_t))$$
$$where, \kappa_t(n) \rightarrow sinusoidal\ kernel\ of\ frame\ 'n'$$
$$W(n) \rightarrow Window\ function\ (Hann\ window\ is\ used)$$
$$\tau_t \rightarrow Frequency\ that\ maximizes\ the\ spectrum\ in\ tempogram$$
$$\varphi_t \rightarrow Phase\ of\ kernel\ dervied\ above$$

As seen above the kernel is windowed. This is done to smoothen it. As the last step, the kernels across frames are aggregated and half-wave rectified to obtain the robust PLP representation.

3.5 Database Creation

After the development of the algorithms used to evaluate the singers, the next step in the implementation of the software was the creation of a database. This database would have a wide range of scores to choose and practice from. As mentioned before, the online music notation software, noteflight, was used to build the database of scores to practice.

The scores were created such that both a professional singer as well as a novice singer could benefit from it. Exercises were designed using a combination of Indian and Western teaching methods. Scores ranged from simple one note sequences, where the challenge is to be able to accurately hit a particular not a and hold it to more complex sequences that had melodies attached to them. Examples of the scores are shown below. The author composed most of the scores in the database primarily drawing from his knowledge of Indian teaching methods and by

consulting experts in Western methods of teaching. The scores that weren't composed by the author were popular tunes such as "Happy Birthday to you", "Twinkle twinkle little star" etc.

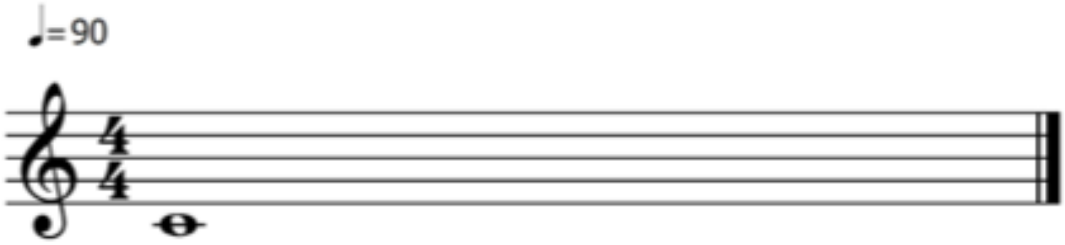


Figure 3.5: Sample of one-note sequence (C4)

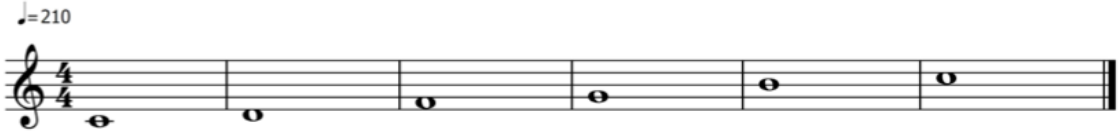


Figure 3.6: Six-note sequence

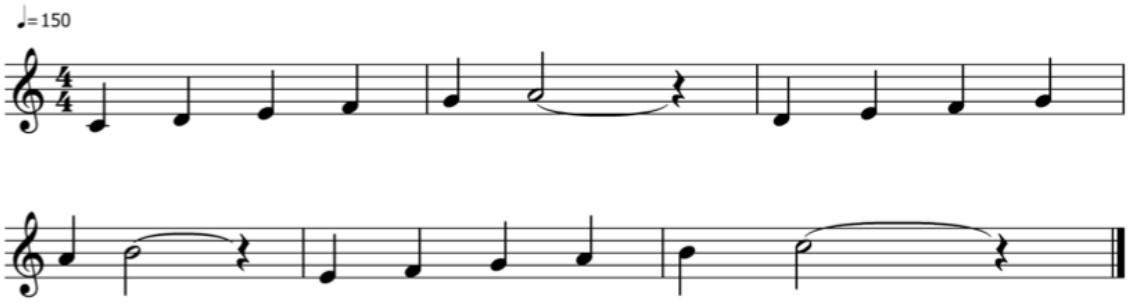


Figure 3.7: Complex Sequence

The simple one-note sequences help the novices practice one-note at a time and get comfortable in hitting a particular note and holding it. This exercise, however, is also helpful for a more experienced singer if they tend to find difficulty in hitting a particular note in a piece. Using this, they can practice it in isolation before attempting to sing the complex tune.

The exercises are designed such that it is semi guided. The sequences are grouped into the number of notes present in the score (for example, figure 3.2 comes under the one-note sequence group while figure 3.3 comes under six-note sequences and figure 3.4 is a complex sequence – more than 7 notes). A person can go through the database in steps one-note to three-notes to five notes and so on or they could skips levels if they are comfortable. This flexibility makes the software robust and semi-guided in its teaching.

The database created had the score in three formats, namely pdf, wav and midi. The pdf version was to view the score on the screen. Images of the same are shown above. The wav file was created for playback. Playback is possible under two conditions. One can listen to the track and familiarize before recording or to compare the performance with the actual file. The midi version was created to send to the score follower as it performs midi-audio alignment.

Currently, the database houses 136 scores divided into 8 groups in the following way:

- One-note sequences: 22 scores
- Three-note sequences: 18 scores
- Four-note sequences: 26 scores
- Five-note sequences: 22 scores
- Six-note sequences: 20 scores
- Seven-note sequences: 8 scores
- Seven-note sequences: 8 scores

- Complex sequences: 12 scores

3.6 Max Patch – UI & Link to pitch and tempo algorithms

The Max/MSP software is used for data acquisition, the user interface (UI) and the as a connection tool to the pitch and tempo algorithms implemented in matlab. The Max patch can be divided into three sections:

i. The Data Acquisition Section – In this section, the user is able to select and load a score from the database. He/she can then listen to the sequence, practice until comfortable and record it.

ii. The Evaluation Section – The user here selects the score he wants to evaluate. The patch connects with the matlab functions in this section. The patch conveys the file path information to matlab, which then retrieves the corresponding files and performs analysis.

iii. The Playback Section – Users here are prompted to select two scores they wish to listen to. One for the left channel and one for the right channel. The pieces selected are then played simultaneously. This is created to be able to listen to performances. One can call this section a form of self-evaluation. One can hear different iterations of their practices against one another or against the actual file in the database and understand how different they sound across performances. This combined with the pitch and tempo information can actually help the singer understand what works and what does not. Often musicians are told that they were technically correct but somehow the end product didn't work. This section can help improve in this respect.

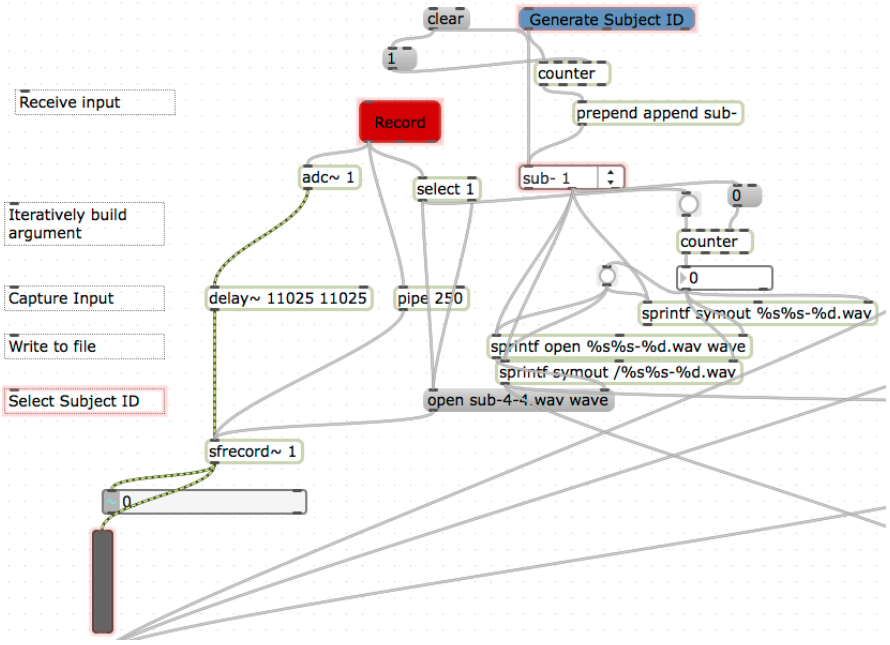


Figure 3.8: The data acquisition section

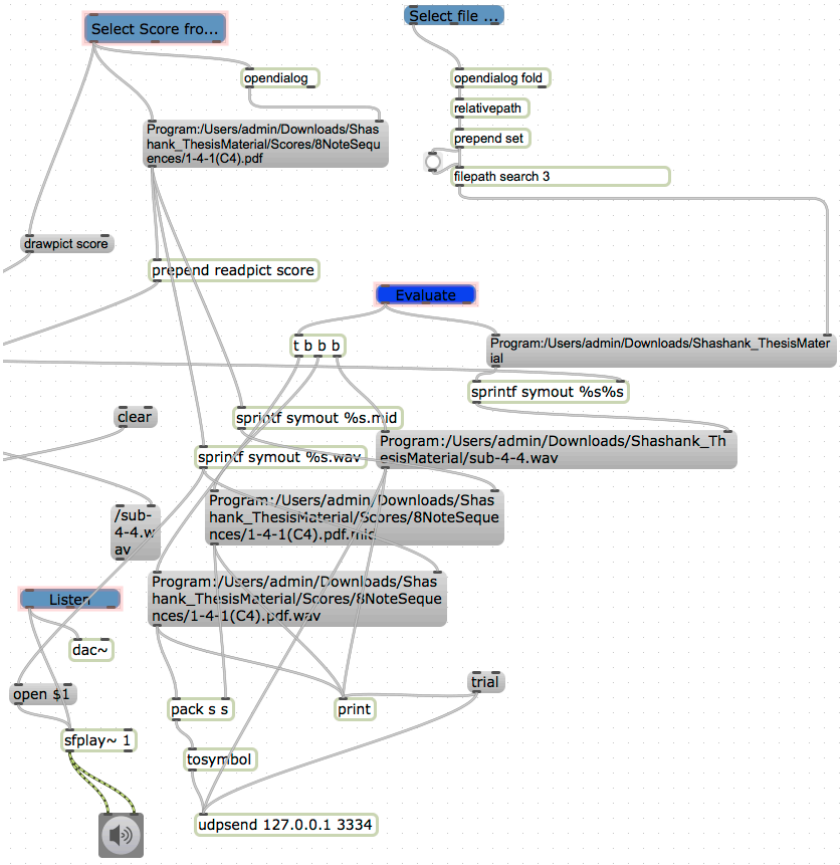


Figure 3.9: The Evaluation Section

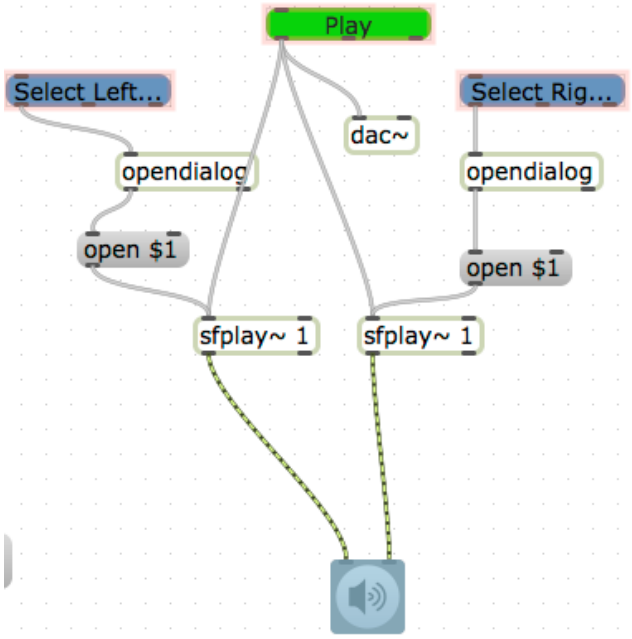


Figure 3.10: The Playback Section

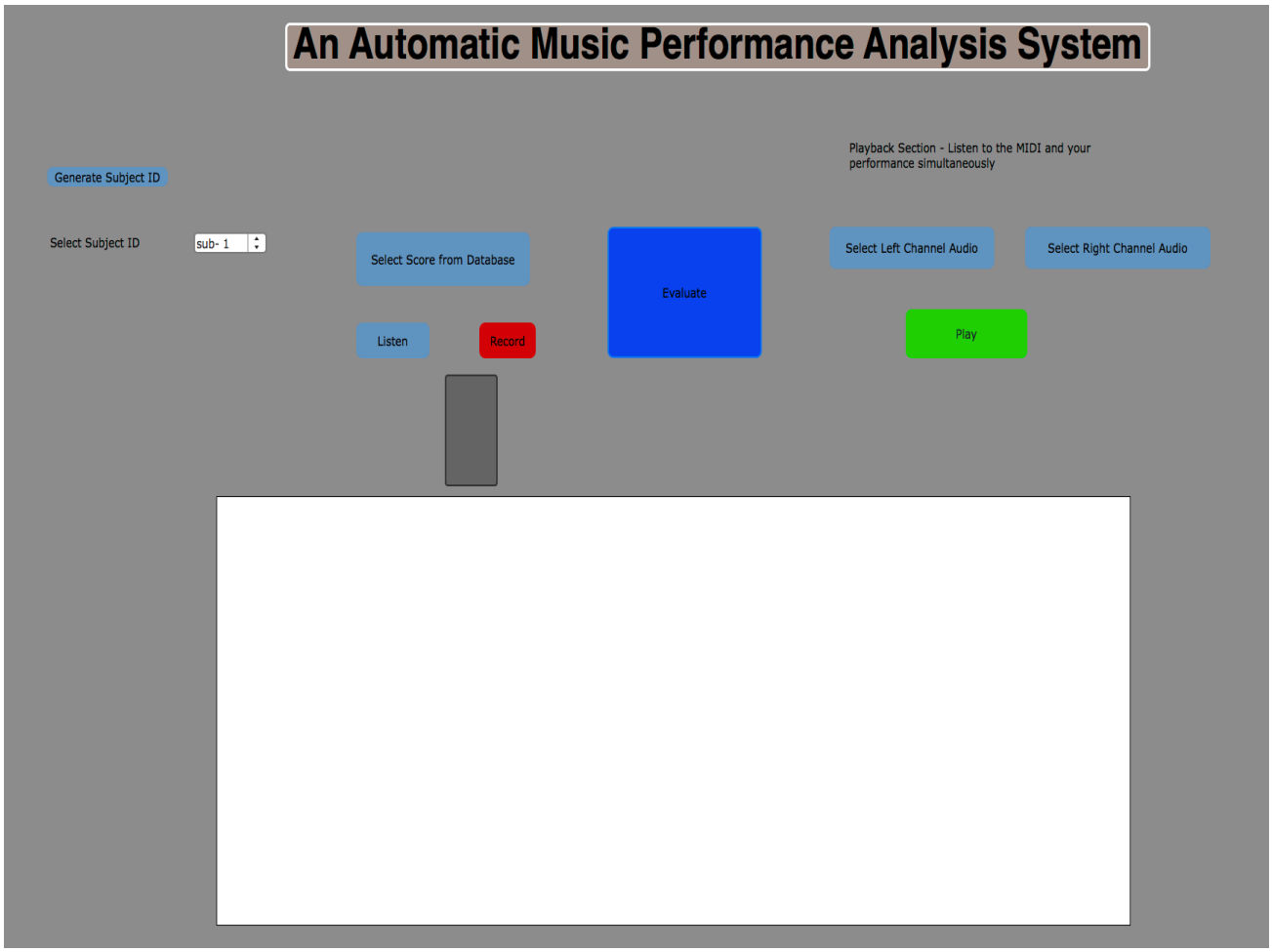


Figure 3.11: The Final User Interface

The figure above shows the final UI of the software. As seen form the figure, this has text buttons and prompts to progress from one section to the next. The prompts for score selection, and audio selection are hyperlinks, which direct the user to the appropriate folder in the system.

3.6.1 Connection between Max and Matlab

A connection between Max and Matlab is possible using UDP (User datagram Protocol) and OSC (Open sound control) messages.

UDP is a flexible protocol used for message exchange between applications. The advantage with this is that it is simple and does not require a connection to be present at all times. Instead, the receiving application watches a port for incoming messages in the form of data packets (datagrams). Although UDP isn't the most robust method to send messages between applications, its simplicity and the simplicity of the problem at hand make it an ideal platform in this project.

OSC is a format in which messages are transmitted and received between applications. Here, Max needs to send messages with file path information to matlab. The udpsend and udpreceive objects in max are used for this purpose. udpsend serializes Max messages into OSC compatible UDP datagrams. At the matlab end, this datagram is read in as an OSC message and the relevant information is extracted from the struct. When matlab sends an OSC message over UDP, the udpreceive object in Max decodes is into a Max message.

The problem with this communication setup is when multiple messages have to be sent. In this case, the path information of three files (the recorded file, the corresponding score file in .wav and .mid formats) needs to be sent to matlab. Timing issues arise with matlab receiving the messages in different orders with no specific pattern. The solution for this problem is to delay the messages or for more robust cases, use handshaking between Max and matlab to send and decode messages one at a time. On the flipside, however, both solutions slow down the evaluation process.

4. Subjective Evaluation

The worthiness of software can be deemed by how helpful the it is, and most importantly how easy is it to use? The first question was answered in the previous sections, wherein it was shown that there is a lack of such a tool for singers. Having successfully tackled that, the natural step forward is to evaluate the software by the other two metrics. A good method to judge its quality is to recruit subjects to test out the software and give feedback.

4.1 Participants

Subjects were carefully chosen for this study so that they came under varied groups like professional singers with and without formal training, novice and enthusiastic singers. This enabled the author to collect opinions of different kinds and thus make the exercise more meaningful. To elaborate, the needs of a professional singer might be different from a novice or a person who just wishes to sing as a hobby. Collecting their feedback on the drawbacks of the system will enable the final implementation of the software a more robust one and can cater to a larger audience.

Thus five graduate and doctoral students (3 male, 2 female) of New York University, most of whom had a musical background were recruited for the experiment. Their mean age was 28.8 years (SD = 5.81). Subjects had an average of 2.2 years of formal training in singing (SD = 2.17) and gave themselves an average self rank in singing skill level of 3.3 (SD = 0.45) on a scale of 1-5. The mean

self rank of overall musicianship ability was 4.1 (SD = 1.02) and one subject reported having absolute pitch.

4.2 Experiment Setup

The experiment was conducted in the Spatial Audio Research Laboratory in the Music Technology department at New York University. This room was specifically chosen, as it is a sound isolated and semi-anechoic. Thus it was ensured that the recordings done were free from external noise or unwanted, disturbing reverberation. The experiment was run on a Mac Pro machine and the interface was seen on a 22' Samsung monitor. The recordings were collected via a Shure SM58 microphone. Playback option was provided either over Sennheiser HD650 headphones or over Genelec speakers.

4.3 Procedure

Participants were seated on a chair about two feet from the computer screen. Before the start of the experiment, they were given the informed consent form and were informed that the University Committee on Activities Involving Human Subjects of New York University approved the experiment being conducted. Following this, the subjects filled a short background questionnaire, which asked them general information about their musicianship and some demographic information. See Appendix A for detailed information about the questions asked of the subjects.

As the next step, the participants were briefed about the software and its features and the tasks to be performed by them. For the playback section,

participants were given an option to listen over Sennheiser HD650 headphones or over loudspeakers. All the participants listened to audio over loudspeakers.

The participants were then given some time to get confortable with the interface. No recordings, analysis or feedback was given to them at this stage. Once they indicated they were ready, connection between Max/MSP and Matlab was established and the participant picked a score of his/her choice from the database. Subjects were given the freedom to listen to the score as many times as they liked before recording. During recording phase, however, listening was prohibited. That is, one could not play the audio sequence and record over it. Once a sequence was recorded, the subject received visual feedback from matlab in the form of plots. After going through the feedback, the subject was given three options:

i. Continue practicing on the software. They were given total freedom. In that, they could choose the same score again, or a more/less complex score to perform.
ii. Listen to audio feedback in the form of simultaneous feedback
iii. Stop the experiment and fill up the post experiment survey questionnaire.

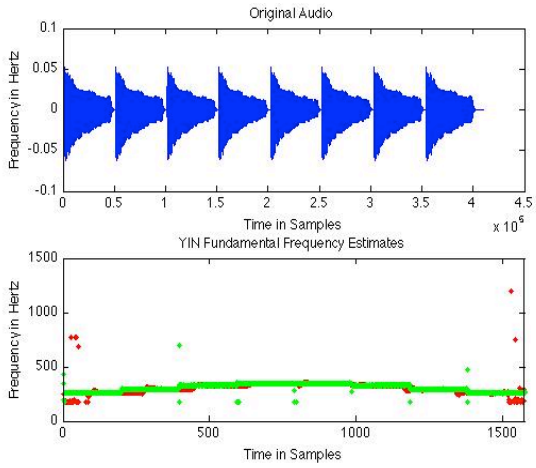
It was observed that, in general, the participants choose to practice with 3-4 different scores before filling up the survey questionnaire. This questionnaire asked them to rate the software against parameters like ease of use, helpfulness of feedback, etc. Refer to Appendix A for the full list of questions.

4.4 Data Analysis

All five subjects were happy with the feedback being provided. Their responses across different criteria can be aggregated and summarized as follows:

- The average rating for the ease of use of the software was 4.6 (SD = 0.55) on a scale of 1-5, where 1 = very complicated and 5 = very easy.
- Four out of the five subjects found the feedback provided helpful and believed it would enhance their practice experience. Two of them, felt that although the feedback was helpful, it more feedback and/or feedback in slightly different formats could make it even more helpful, while two subjects were happy with the format and visualization of the pitch and tempo information.
- 40% of the participants found the Hz scale for feedback non-intuitive and therefore not helpful.
- 60% of the participants (three of the five subjects) said they would use the software regularly during their practice sessions, while one mentioned it would be used occasionally and the last mentioned it would be used every time.
- All five subjects mentioned that they would recommend this software to their peers.
- Lastly, all five subjects were very happy with the simultaneous two-performance playback feature. All of them felt that audio feedback is as helpful as visual and quantitative feedback.

Elaborating on the pitch feedback provided by the software, the following figures show the current method of feedback given.



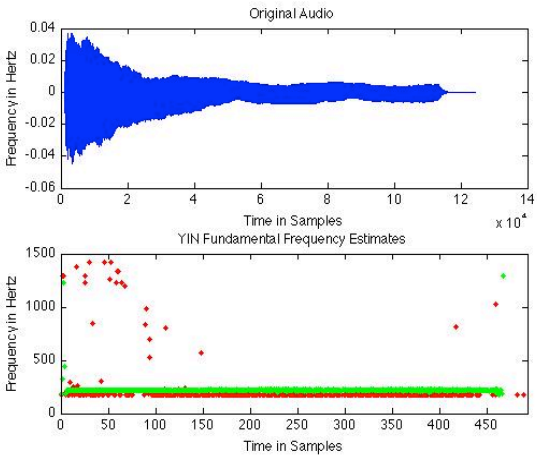


Figure 4.1a: Pitch feedback for a complex sequence Figure 4.1b: Feedback for 1-note sequence

These are zoomed out images of the feedback. The green line represents the expected pitches while the red lines indicate the pitches sung. If a correct note is hit, like a large portion of the sequence in figure 5.1a, the red and green lines overlap with the green one being given prominence to indicate a correct note. The figure also informs the user in Hz, how much off the expected value he/she is. As mentioned above, the Hz scale was found non-intuitive and therefore less helpful by 40% of the users.

5. Discussion

This thesis aimed at providing a novel software for singers to practice in the absence of an instructor. While the techniques used or the type of feedback given have been worked on and implemented before, this type of aggregation as a package is missing. The semi-guided tutoring approach system has also not been attempted or implemented before. Thus the Automatic Music Performance Analysis System took form to fill in the gap. The attempt has been by and large successful.

Using knowledge of Indian and Western music practices for teaching singing, a large database was created for the user to be able to choose from and strengthen one's singing skills from basics. The flexibility and semi-guided nature of the software enabled both professional and novice singers a way to utilize this tool to enhance their practice experience. The small subjective evaluation done for this confirmed the above statement. Subjects testing the software were pleased with its features and 80% of the subjects stated that they would be using this tool on a regular basis in their practice sessions. 100% of the users stated that the large and comprehensive database enabled them to practice in a meaningful and directed manner. The audio playback feature implemented, another novel contribution, was again very well received as this enabled users to actually compare performances simultaneously, and while it did not give any quantitative feedback, it really enhanced their experience as audio feedback along with the visual could help them understand the 'missing element' when even a correct note is sung. 60% of the participants believed this addition made took the software to a new level, and 40% of

the subjects said it was the 'coolest feature and one that is essential but not very intuitive.'

With this all in one package, this software provides an opportunity for singers to practice, get feedback (both audio and visual), choose a score to practice from and store their practices for later reference. Thus a singer can monitor his/her progress over a period of time without relying on expert opinion. All these features open way for a new method of practice that is robust, user friendly and helpful.

While all these positive statements have been made about the software, there has been some criticism as well. 40% of the subjects felt the Hz system to denote the pitch information did not work. It was not an intuitive scale and one that was difficult to understand. It was felt that an error in say 15-20Hz did not mean anything. All that could be understood was whether the sung note was flat or sharp. The "how much" part of the error was not best explained in the Hz scale. It was suggested that MIDI pianoroll might be a more intuitive way to assess the severity of the error. That is to say, "G4" was sung when "A4" was expected makes more intuitive sense than the sung pitch was 392Hz while the expected pitch was 440Hz. The second negative feedback on the same issue was the use of frequency plots to depict the performance. Performance curves again deemed not as intuitive as pianorolls.

Another criticism from 60% of the participants was the lack of a metronome to count them in and help them keep track of the tempo. This feature was omitted by design rather than accident. The Indian methodology of teaching is to attempt to

master the tempo in the absence of a metronome. It is believed that only then does one truly get complete command over tempo. Retrospectively, the author believes that the metronome option would help singers, especially novice ones to get tempo support and assistance from the software if they want it.

Weighing up the criticism and the praises for the software, one can come to the conclusion that the software has been a success as the praises outweigh the criticism. 80% of the participants wish to utilize this tool regularly during practice, it has a mean rating of 4.6 for user friendliness (on a scale of 1-5) and 100% of the subjects stated they would recommend this software to their peers. This shows that the participants felt the flaws existing in the system could be overlooked and it can be recommended to others. Thus one can say the development of this software has been a success.

6. Conclusions and Future Work

6.1 Conclusions

In conclusion, this thesis proposed and built a novel software for singers to practice in the absence of an instructor. The novelty lay in the fact that the software brought in a database of scores to practice from, which were built using the author's Indian teaching methodology and informal expert opinion on Western teaching methodologies. This database enabled the user to utilize the software in a semi-guided manner, which again was novel approach. Lastly, a new feature was added to the software, which gave audio feedback to the user so that he/she could listen, see and learn at the same time. This was found to be very innovative, useful and counter-intuitive feature by some of the participants in the subjective tests that were conducted to evaluate the worthiness of the software. The goals of the thesis were achieved successfully, as proven by the results from the subjective tests, which showed that the participants found the software helpful, easy to use and one that is worthy of a recommendation. While there is still work to be done to make this software better from a helpfulness perspective and on the robustness front, this is definitely a step in the right direction.

6.2 Future Work

The work presented in this thesis is by no means a finished one. There can be several steps taken to improve the software. Some of the possible directions of future research in this topic are:

- Creation of a web-based application using HTML5 or similar technology.

- The database has great scope for improvement. In its current state, the user is restricted to practice of the scores present in the database. Going further, features like importing in a score from the computer/internet, creating a score (music notation capability) can easily be added.
- With a web-based application, one could potentially have the option of adding any score composed or imported in locally to be added to the main server for the benefit of all the users of the software.
- The pitch feedback format can be looked into and implementation of the MIDI pianoroll for those uncomfortable with the HZ system could be implemented.
- As stated in the discussion, the metronome facility is one that could be added especially keeping in mind the primary target audience (novice singers).

As a concluding remark, the author hopes that research and development of this software continues and it one day becomes an integral part of the music community.

References

Bretos, J., and Sundberg, J., (2002). Measurements of vibrato parameters in long sustained crescendo notes as sung by ten sopranos. TMH-QPSR, KTH, Stockholm, Vol. 43, pp. 37-44.

Brown, J., C and Zhang, B., (1991). Musical Frequency tracking using the methods of conventional and narrowed autocorrelation. J. Acoust. Soc. Am. 89 (5).

Cont, A. (2006). Realtime audio to score alignment for polyphonic music instrument using sparse non-negative constraints and hierarchical HMMs. IEEE, ICASSP.

Cont, A. (2010). A coupled duration-focused architecture for real-time music-to-score alignment. IEEE transactions on pattern analysis and machine intelligence. Vol 32. No. 6.

Cont, A. (2009). A Coupled Duration-Focused Architecture for Real-Time Music-to-Score Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(6), 974–987.

Dannenberg, R. B. & Ning Hu. (2003) "Polyphonic Audio Matching for Score Following and Intelligent Audio Editors." Proceedings of the 2003 International Computer Music Conference. 27-33.

Davies, M., E., P., and Plumbley, M., D., (2005). Comparing mid-level representations for audio based beat tracking.

Davies, M., E., P., and Plumbley, M., D., (2005). Beat tracking with a two state model. Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). pp. 241-244.

De Cheveigné, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, *111*(4), 1917.

Devaney, J., and Ellis, D., P., W., (2009). Handling asynchrony in audio score alignment. Proceedings of the International Computer Music Conference.

Devaney, J., Mandel, M., and Fujinaga, I., (2011) A study of intonation in three parts singing using the automatic music performance analysis and comparison toolkit (AMPACT). International Society for Music Information Retrieval.

Devaney, J., Mandel, M., and Ellis, D, P., W., (2009). Improving MIDI-audio alignment with acoustic features. IEEE workshop on applications of signal processing to audio and acoustics.

Devaney, J., Mandel, M. I., Ellis, D. P. W., & Fujinaga, I. (2011). Automatically extracting performance data from recordings of trained singers. *Psychomusicology: Music, Mind and Brain*, *21*(1-2), 108–136.

Devaney, J., Mandel, M., and Fujinaga, I., (2011). Characterising singing voice fundamental frequency trajectories. IEEE workshop on applications of signal processing to audio and acoustics.

Dixon, S., (2005). Live tracking of Musical performances using on-line time warping. Proc. of 8th International Conference on Digital Audio Effects (DAfx'05).

Dixon, S. (2005). An On-Line Time Warping Algorithm for tracking musical performances. Proceedings of IJCAI 2005, 1727-1728

Dixon, S. (2005) Live tracking of musical performances using on-line time warping. Proceedings of the 8th International Conference on Digital Audio Effects, 2005

Ellis, D., P., W., (2007). Beat tracking by dynamic programming. From the wiki-page of the MIREX retrieved from http://www.music-ir.org/evaluation/

Ewert, S., Muller, M., Grosche, P., (2009). High resolution audio synchronization using chroma onset features. IEEE transactions, ICASSP 2009.

Ewert, S., Mueller, M., and Grosche, P. (2009) "High resolution audio synchronization using chroma onset features," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Taipei, Taiwan, Apr. 2009, pp. 1869–1872.

Gockel, H., Moore, B. C. J., & Carlyon, R. P. (2001a). Influence of rate of change of frequency on the overall pitch of frequency-modulated tones. *The Journal of the Acoustical Society of America*, *109*(2), 701–712. doi:10.1121/1.1342073

Gockel, H., Moore, B. C. J., & Carlyon, R. P. (2001b). Influence of rate of change of frequency on the overall pitch of frequency-modulated tones. *The Journal of the Acoustical Society of America*, *109*(2), 701–712. doi:10.1121/1.1342073

Grosche, P., and Muller., M (2011). Extracting predominant local pulse information from music recordings. IEEE transactions on Audio, Speech, and Language Processing. Vol. 19, No. 6.

Grosche, P., Muller, M., and Kurth, F., (2010). Cyclic tempogram – A mid-level tempo representation for music signals. Proceedings of the 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). pp. 5522-5525.

Grosche, P., and Muller, M., (2009) A mid-level representation for capturing dominant tempo and pulse information in music recordings. International Society for Music Information Retrieval.

Grosche, P., and Muller, M., (2009). Computing predominant local periodicity information in musical recordings. IEEE workshop on applications of signal processing to audio and acoustics.

Grosche, P., & Muller, M. (n.d.). Extracting Predominant Local Pulse Information From Music Recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(6), 1688–1701. doi:10.1109/TASL.2010.2096216

Heller, J., J., and Campbell, W., C., (1971) Musical Performance Analysis. Bulletin of the council for research in music education No. 24, pp. 1-9.

Hochenbaum, J., and Kapur, A. (). Towards improving onset detection accuracy in non-percussive sounds using multimodal fusion.

Klapuri, A., P., Eronen, A., J., and Astola, J., T., (2004) Analysis of the meter of Acoustic Musical Signals. IEEE transactions on Speech, and Audio Processing.

Koblyakov, L. (1992). Score/Music Orientation: An interview with Robert Rowe. Computer Music Journal. Vol. 16, No. 3, pp. 22-32.

Lahat, M., Niederjohn, R., J., Krubsach, D., A., (1987). A spectral autocorrelation method for measurement of fundamental frequency of noise-corrupted speech. IEEE transactions on Acoustics, Speech, and Signal Processing, Vol. 35, No. 6.

Macon, M., W., Link, L., J., Oliverio, J., Clements, M., A., and George, E., B., (1997). A singing voice synthesis system based on sinusoidal modeling. Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). pp. 435-438

Mandel, M., Ellis, D, P., W., and Fujinaga, I., (2011). Automatically extracting performance data from recordings of trained singers. Psychomusicology: Music, Mind and Brain. Vol 21, No. 1 & No. 2.

McKinney, M. F., Moelants, D., Davies, M. E. P., & Klapuri, A. (2007). Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms. *Journal of New Music Research*, *36*(1), 1–16. doi:10.1080/09298210701653252

Mueller, M., Kurth, F., and Clausen, M. (2005) Audio matching via chroma-based statistical features. In Proc. Int. Conf. on Music Info. Retr. ISMIR-05, pages 288–295.

Mueller, M. (2007) Information Retrieval for Music and Motion. Springer, 2007.

Orio, N. & F. D´echelle (2001). Score Following Using Spectral Analysis and Hidden Markov Models. Proceedings of the 2001 International Computer Music Conference.

Palmer, C. (1997). Music Performance. Annu. Rev. Psychol. 48, 115–138.

Prame, E., (1992). Measurements of the vibrato rate of ten singers. Journal of the

Acoustical Society of America 94: 1979–84.

Prame, E., (1997). Vibrato extent and intonation in professional western lyrics singing. J. .Acoust. Soc. Am. 102(1).

Puckette, M., & Lippe, C. Score Following in Practice. In Proceedings of the International Computer Music Conference, 182-185.

Puckette, M. (1995). Score following using a sung voice. Proceedings of the International Computer Music Conference, 199-206.

Raphael, C. (2002). Automatic Transcription of Piano Music. Proc. Int. Conf. Music Inf. Retrieval., 2002, 15–19.

Repp, B. H. (1992). Diversity and commonality in music performance: An analysis of timing microstructure in Schumann's ''Träumerei''. J. Acoust. Soc. Am. 92(5), 2546-2568

Schwarz, D., Orio, N., Schnell, N. (2004). Robust Polyphonic Midi Score Following with Hidden Markov Models. Proceedings of the International Computer Music Conference.

Appendix A

Background Questionnaire
An Automatic Music Performance Analysis System

Subject Code:

Age:

Gender:

1. Do you sing professionally and/or how many years of formal training (i.e. lessons) do you have?

2. How would you rank your singing skills on a scale from 1 to 5? (1 is worst, 5 is best)

3. Other musical training in number of years

Instrument(s):

Composition:

Other (please specify type as well as number of years):

4. Rank your overall level of musical training from 0 (no musical training) to 5 (professional):

5. What kinds of music do you listen to? If multiple kinds, list in order of preference.

6. How often do you listen to music? (Indicate in hours/day)

7. Do you have absolute pitch, also known as perfect pitch (circle one)? YES NO Don't Know

8. Anything else about your musical background you'd like to mention?

Survey Questions
An Automatic Music Performance Analysis System

Subject Code:

1. Do you find the feedback provided on your performance helpful (circle one)? YES NO

2. If No, what is lacking in the feedback?

3. Rank the ease of use of the software from 1 (Very Difficult) to 5 (Very Easy)

4. If this software was available, how often will you use it?

- Every time I practice
- Regularly, but not every time
- Occasionally
- Never

5. Would you recommend this software to your friends? YES NO

6. Anything else you'd like to mention about the experiment?