A Stochastic Method of Tracking a Vocal Performer

Lorin Grubb and Roger B. Dannenberg
School of Computer Science, Carnegie Mellon University
{lgrubb, rbd}@cs.cmu.edu

Abstract

Automated accompaniment systems are computer systems that "play along" with a solo musician or a group of musicians when given the score for a piece of music. These systems must be able to "listen" to the live musicians by tracking their progress through the score in real-time. Accurate tracking of vocal performers (as opposed to instrumentalists) is a particularly challenging case. In this paper, we present a tracking method based upon a statistical model of vocal performances. This technique incorporates both information obtained from real-time signal processing of a performance (such as fundamental pitch) and information describing the performer's movement through the score (namely tempo and elapsed time). We present a description of how this model is incorporated as part of a system currently used to accompany vocal performers. We provide a preliminary evaluation of its ability to estimate score position of a performer.

1 Introduction

Automated accompaniment systems are computer systems designed to accept a musical score as input and to provide real-time performance of the accompaniment in synchrony with one or more live soloists. Automated accompaniment systems must concurrently execute several tasks within the real-time constraints of musical performance. First, these systems must observe the soloists by detecting what they have performed. If the soloists' performances do not involve electronic instruments, this will likely require some form of audio signal processing to extract relevant *features*, such as fundamental pitch. Second, accompaniment systems must track the soloists as they perform the score. Tracking often involves both identifying the soloist's current score position and estimating the soloist's tempo. Third, the systems must react to the soloists by tastefully performing the accompaniment, generally attempting to synchronize with the live performers. Finally, accompaniment systems must generate the actual sound for the accompaniment. Sound production is usually accomplished either by controlling audio synthesizers or by directly generating digital audio.

Several systems for accompanying a vocal performer have been previously described [5] [3] [4] [7]. The first three systems accompany amateur vocalists performing pop music. The first two rely on pitch detection for tracking the performer, while the third applies speech processing techniques for vowel recognition. These systems attempt to identify both the score position and the tempo of the performer, and to adjust the computer accompaniment in response. The fourth system was used to accompany a contemporary art piece written for computer and soprano. It relied on pitch detection and did not attempt to determine tempo of the performer. Rather, it was designed for fast identification of soloist notes that were scored to coincide with computer generated sounds.

The designers of these systems commonly report certain problems that complicate tracking of a vocalist. These include variation of detected features (such as pitch) resulting from accidental as well as intentional actions on the part of performers. In addition, methods for pitch detection and vowel detection are generally not themselves error-free. Consequently, all of these systems incorporate heuristics or weighting schemes intended to compensate for mistakes made when features are directly matched against the score.

We present a technique for tracking a vocalist that is based upon a stochastic description of a performer's score position. It incorporates a variety of relevant information including recent tempo estimates, features extracted from the performance, and elapsed time. Unlike previous systems, it does not require subjective weighting schemes or heuristics. It can use either formally derived or empirically estimated probabilities describing the variation of the detected features and other relevant data.

In addition, we describe how this score following model can be efficiently implemented on even low-end PC's, so as to satisfy the real-time constraints imposed by musical accompaniment. We describe how this technique is currently used as part of a system to accompany vocal performances, providing some

preliminary measurements of its position estimation ability. We conclude with a discussion of some future work to enhance the statistical model and possibly lead to improved vocal tracking.

2 A Stochastic Model of Location

As previously mentioned, a musical score will often consist of one or more solo parts and an accompaniment. In the case of Western classical music written for a single vocalist, the solo part will consist of a sequence of notes, each note indicating at least pitch, a syllable to be sung, and relative duration. Other information, such as dynamic and articulation, may also be specified. Also, the tempo for a given piece will likely vary within a single performance, as well as across performances. Tempo variations may be explicitly written in the score by the composer, or may be the result of conscious choices made by the performer.

The model we use to track a vocalist represents the vocalist's part as a sequence of *events* that have a fixed, or at least a desired, ordering. Each *event* is specified by:

1. A *relative length* which defines the size or duration of the event, as indicated in the score, relative to other events in the score.
2. An *observation distribution* which completely specifies the probability of every possible sensor output at any time during the event.

The relative length may be specified in beats for a fixed tempo, or in some unit of time resulting from the conversion of beats to "idealized time" using a fixed, idealized tempo. The length is assumed to be real-valued and not necessarily a positive integer.

The vocalist's part in the score is thus viewed as a sequence of events, each event spanning a region of a number line. The *score position* of a singer is represented as a real number assuming a value between 0 and the sum of the lengths of all events in the score. Score position is thus specified in either idealized beats or idealized time, and can indicate the performer's location at a granularity finer than an event.

At any point while tracking an actual performance, the position of the vocalist is represented stochastically as a continuous density function over score position. This is referred to as the *score position density*. The area under this function between two score positions indicates the probability that the performer is within that region of the score. This is depicted in Figure 1. The area over the entire length of the score is always 1, indicating it is 100% likely that the performer is in the score. As the performance progresses and subsequent *observations* (detected features) are reported, the score position density is updated to yield a probability distribution describing the performer's new location.

The observation distribution for each event specifies the probability of observing any possible value of a detected feature when the vocalist is performing that event. This distribution will generally be conditioned on information provided in the score. For example, if pitch detection is applied to the performance, then the observation distribution for a given event might specify for each pitch the likelihood that the detector will report that pitch, conditioned on the pitch written in the score for that event. As another example, distributions might also describe the likelihood of detectable spectral features that are correlated with sung phonemes.

Our approach to tracking the performer is conceptually simple. For each new observation, we use the current score position density and the observation distributions to estimate a new score position density. This updated density indicates the current location of the performer in the score. In practice, calculating a new score position density requires a number of simplifications, assumptions, and approximations. In order to describe our system, we will first present a mathematical model for updating the score position density. Then we will describe an implementation of this model.

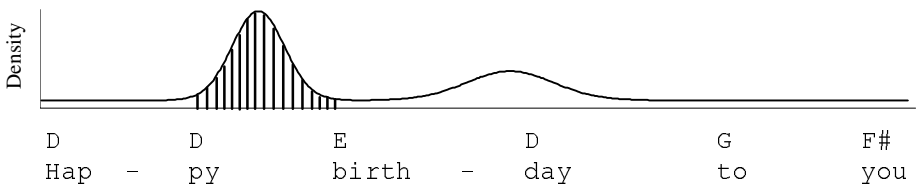

Figure 1: Example of a density function characterizing the score position of a performer. The area of the shaded region gives the probability that the performer is singing the second note.

The model for updating the score position density incorporates three pieces of information that are relevant to determining the new position of the performer, referred to as the performer's *destination position*. First, since a performer's rendering of a musical score is highly sequential, it is important to consider the performer's location at the time of the previous observation. This location will be referred to as the performer's *source position*. Second, the observation most recently extracted from the performance will obviously provide information about the performer's current location. Finally, performers often attempt to maintain a consistent tempo, subject to relatively minor and gradual variations. An estimate of the performer's tempo in the recent past, along with the elapsed time since the score position density was last updated, can give a useful prediction of how much score was performed during that elapsed time. This prediction will be referred to as the *estimated distance traversed*, or simply the *estimated distance*.

Given these three variables—previous position, most recent observation, and estimated score distance traversed by the performer—the current location of the performer can be specified stochastically by the following conditional probability density:

$$f_{I \mid D,V,J}\,(i \mid d, v, j)$$

i = the performer's destination position
d = the estimated distance
v = the observation
j = the performer's source position

Unfortunately, directly defining this multidimensional function for each and every score would be very challenging. Also, the previous score position of the performer is never known with certainty, so the value of at least one conditioning variable, j, should also be described stochastically. This can be accommodated by performing the following integration:

$$f_{I \mid D,V}\,(i \mid d, v) = \int_{j=0}^{\|Score\|} f_{I \mid D,V,J}\,(i \mid d, v, j) \cdot f_{J \mid D,V}\,(j \mid d, v)\,\partial j$$

where $\|Score\|$ represents the length of the score. Note that additional integration would be required if the values of the estimated distance, d, and observation, v, were also specified stochastically.

While this formulation is a good starting point and very comprehensive, it is impractical for direct implementation. Since some of the functions in the integral are likely to be specified numerically, a closed-form solution is not possible. Also, the density functions are conditioned on so many parameters that estimating them from real data would require a large number of observations. In the subsequent discussion, we introduce a number of approximations and simplifications that transform the original model into one that is both practical and effective. The following simplifying assumptions are made:

1. The estimated distance, d, and the observation, v, are not specified stochastically as distributions, but are reported as scalar values produced by tempo estimation and signal processing algorithms, respectively. This reduces the dimensionality of the model, thus simplifying each update of the score position density.
2. The observation, v, depends only on the destination position, i, and is independent of both the performer's previous score position, j, and the estimate of the score distance, d. This assumption is certainly not completely accurate. However, to the extent that the performer renders the score in a highly sequential fashion and the model updates occur frequently enough so that d always assumes a value within a small range, this simplification is likely to be reasonable.
3. Under assumption 2, $f_{J \mid D,V} = f_{J \mid D}$. We further assume that the score position density resulting from the previous model update is a reasonable approximation to $f_{J \mid D}$ for the given value of d. Thus we substitute the previous estimate of the performer's location, $f_{Source}(j)$, for $f_{J \mid D,V}(j \mid d,v)$ in the previous integral.
4. A distribution describing the actual amount of score performed by the vocalist between updates of the score position density is independent of the performer's source location. It only depends on the estimated score distance, d. This allows the performer's motion through the score to be modeled as a *convolution integral*.

While none of these assumptions is completely accurate, in combination they yield a reasonable approximation to the general score following model. This simplified model can be more easily specified and permits for a more efficient computer implementation.

Under the four stated assumptions, the model for score following can be decomposed into two parts. First, an estimate of current location based on prior location and estimated distance can be calculated:

$$f_{I \mid D}\,(i \mid d) = \int_{j=0}^{\|Score\|} f_{I-J \mid D}\,(i-j \mid d) \cdot f_{Source}(j)\,\partial j$$

Next, this estimate can be modified to account for the most recent observation:

$$f_{I \mid D,V}(i \mid d, v) = \frac{f_{V \mid I}(v \mid i) \cdot f_{I \mid D}(i \mid d)}{\int_{k=0}^{\|Score\|} f_{V \mid I}(v \mid k) \cdot f_{I \mid D}(k \mid d)\, \partial k}$$

The result is a score position density conditioned on both the estimated distance and the most recent observation. Note that if d and v represent fixed values (as previously assumed), the result is a one-dimensional function over score position.

The following density functions are assumed to be pre-defined prior to each application of the model:

1. f_{Source} - The stochastic estimate of the performer's source position. Under assumption 3 above, this is the score position density calculated at the time of the previous observation.
2. $f_{I\text{-}J \mid D}$ - The probability that the performer has actually performed an amount of score I-J given D, a prediction of the amount of score performed.
3. $f_{V \mid I}$ - The probability of making observation V when the performer is at position I. This function is specified by the observation distributions of the events that form the score.

The second and third functions can each be defined using one of three alternative methods. First, one can simply rely on intuition and experience regarding vocal performances, and estimate a density function that seems reasonable. Alternatively, one can conduct empirical investigations of actual vocal performances to obtain numerical estimates of these densities. Pursuing this further, one might actually attempt to model such data as continuous density functions whose parameters vary according to the conditioning variables. Theoretical descriptions of performance might be applicable in this case.

3 Efficient Model Implementation

Direct execution of the simplified score following model requires the evaluation of two integrals. To permit for the widest range of possible density functions, we implement the model numerically. The density functions are sampled (i.e., represented in point-value form) and the integrals approximated numerically. Since the first integral in the simplified model contains at least one function with two free variables, direct calculation of this integral would require time quadratic in the number of samples spanning the length of the score.

Fortunately, the first integral is a convolution integral. Numerical evaluation of this integral can be expedited through application of the discrete Fourier transform (DFT). It is a property of this transform that *discrete convolution*, as results from numerical representation of the functions, can be calculated by first computing the discrete transforms of each function in the integral, calculating the product of these transforms, and then applying the inverse of the discrete transform to that product. For a transform of size N, this sequence of operations can be accomplished in time $\mathcal{O}(N \log N)$. For calculating convolutions of even moderately large numbers of samples (certainly $N \geq 100$), this technique is noticeably faster than the direct approach.

Figure 2 provides a flowchart showing the various steps in a single application of the score following model. Also shown is each step's complexity relative to the number of samples, S, along the score position dimension. Note that allowance is made for real-time generation (sampling) of both the distance density function, $f_{I\text{-}J \mid D}$ and the observation density, $f_{V \mid I}$. Computation of the Fourier transforms is the most cumbersome part of the process. Also, convolution via the DFT may require calculating transforms with as many as twice the number of points as the number of samples in the individual functions. This fact is reflected in the complexities shown in Figure 2.

To achieve a tractable implementation, the score position density function is not calculated over the entire length of the score. Instead, the function is calculated over only a portion of the score, referred to as a *window*. Windowing of a score is a technique commonly used to implement automated accompaniment systems, and was first presented by Dannenberg [1]. For purposes of the stochastic model presented here, the score position density is either assumed to be zero outside of the window, or to be sufficiently close to zero as to be of no significance.

Each application of the model can produce an estimate of the score position density for a shifted window, encompassing a region slightly to the left or right of the previous window. Each update uses only those points of the score position density function that are contained within the window from the previous application of the model. The size and direction of the shift can be based upon changes, from window to window, in the region or regions of highest density. Thus the window will essentially move through the score over time, following the performer.

As a low-end test of this implementation, we have executed the model on a PC using an 80486 processor at a clock speed of 66 MHz. Using double-precision floating point and DFT's with 512 points, one application of the model requires 35 ms of CPU time. Since the complexity of calculating the model is nearly linear in the size of the transform, a computing platform which is twice as fast permits a window encompassing twice as many points to be calculated in nearly the same time. Modern processors have enough power to extract features from an audio signal in addition to applying the model. The accompaniment can be generated using a sound card or external synthesizer for a complete accompaniment system.

4 Modeling Vocal Performances

Both the distance and observation density functions must be explicitly defined. To accomplish this, we recorded performances given by live vocalists singing with live accompanists. The vocal performances were recorded in isolation, using a highly directional microphone placed in close proximity to the singer. The recordings can thus be analyzed for both pitch content and tempo. The vocalists had at least 1 full year of university level training. They performed pieces of Western classical music that either were familiar to them or were part of their current program of study.

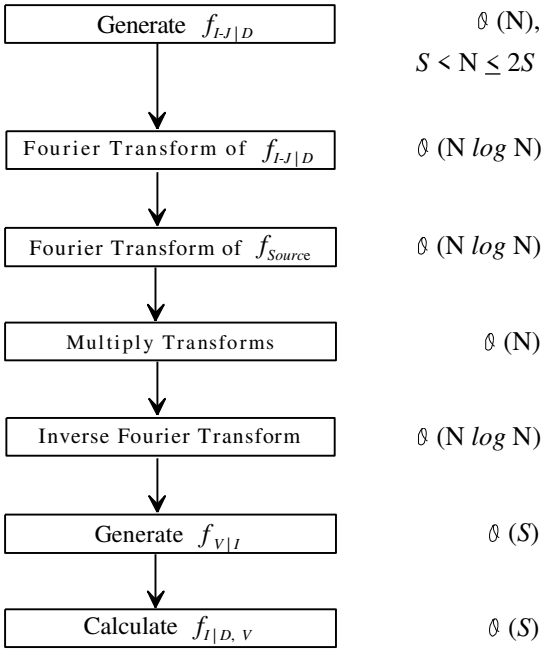

Figure 2: Flowchart for implementing the simplified score following model by using the discrete Fourier transform. Computational complexity is also given, where S represents the number of samples along the score position dimension.

While many relevant features can be generated from a digitized waveform of a vocal performance, our initial implementation has focused only on fundamental pitch. The events for each musical score have an observation density that is conditioned on the pitch that is notated in the score. Thus at present, all events that correspond to A-440 are associated with the same observation distribution. The relative length of the events is based upon an idealized tempo. Tempo changes that are explicitly marked in the score are used to calculate changes in the idealized tempo for different sections of score.

Definition of the density function $f_{I\text{-}J|D}$ will depend on how the estimated distance, *d*, is generated. Our approach is to compute the product of the most recent estimate of the performer's tempo (as used to control the accompaniment) and the elapsed time since the previous update of the score position density. The distance density is conditioned on tempo and elapsed time. It changes for successive calculations of the score following model. Thus to some degree, the score position density reflects changes in both the performer's tempo and the elapsed time between successive observations.

A subset of 20 performances was used to determine empirical density functions. Only 18 of these were used for the distribution of actual pitch conditioned on the pitch in the score, since 2 of the selected recordings contained a low-level background hum that interfered with pitch detection. These recordings contained 2 performances by each of 9 singers encompassing all primary voice types and performing a total of 16 different compositions. All 20 performances were used to estimate the distribution of the actual amount of score performed conditioned on the estimated distance. These recordings contained 2 performances by each of 10 singers, again encompassing all primary voice types and performing a total of 16 different compositions. We believe the resulting empirical density functions to be fair approximations to the respective distributions in the limit for a target population of performances of classical music given by trained singers.

The pitch detection algorithm is based on one described by Kuhn [6]. It uses a bank of lowpass filters spaced at half-octave intervals along the range of the vocalist's part. Bass boost is applied to an analog

audio signal via an external mixer. The audio is digitized at 15KHz by a PC sound card and analyzed in 33 ms blocks. Level control is applied to the blocks prior to filtering. The output of each filter is sent through a zero-crossing detector to determine average pitch period. Maximum amplitude is also determined. Average fundamental pitch for a block is taken from the filter with lowest cutoff frequency whose maximum amplitude exceeds 25% of the maximum amplitude over all filters.

A preset amplitude threshold is used to distinguish pitched signal of interest from blocks containing silence, breathing, consonants in the singing, and low-level background noise. The detector reports the median pitch over every 3 consecutive blocks of pitched signal. Thus during a sustained tone, the detector reports pitch at a rate of 10 Hz.

The 20 recorded performances were played from a DAT tape and processed by the pitch detector. The output was parsed by hand in order to time align the reported pitches with the notes in the scores. This parsing process relied on information about silences and pitch in the detector output, as well as occasional graphical examination of digitized waveforms of the recordings. Next, the distance (in semitones) between the detected pitch and the scored pitch was calculated for the 18 performances. This provided an observation distribution for actual pitch given a scored pitch.

Similarly, the distance density was modeled using all 20 performances. Using the time aligned parses of the pitch output, the performer's tempo was calculated over short, consecutive regions of score. This data was used to model the distribution of the subsequent score distance performed given a previous short-term tempo and a known elapsed time. In contrast to the model for pitch, the distance density is continuous and based upon convolution of lognormal density functions. Further details of this process are beyond the scope of this paper.

5 Accompanying Singers

Several issues must be addressed in order to use the score following model within an actual automated accompaniment system. First, there is a need for a precise interpretation of probability as computed by the model. For purposes of a general accompaniment system, we view the probability specified by the score position density as a frequency count. More specifically, the probability over a region of score indicates the relative number of performances from a target population of performances which, having produced the sequence of observations and tempo estimates so far generated, will find the performer within that region. Thus the purpose of statistical modeling, in both the theoretical and empirical aspects, is to identify a tractable model which closely approximates the actual position distribution among the target population of performances.

Next, since an accompaniment system must control the performance of the accompaniment, we need a method of using the stochastic description of the vocalist's score position to select an accompaniment control action. One possibility is to apply a decision-theoretic approach. This requires the definition of a *loss function*. For every possible position of the performer, this function would quantify the relative, negative impact of taking a particular accompaniment control action. The probabilistic description of a vocalist's position could be used in combination with the loss function to determine an action that would probabilistically minimize the expected loss (negative impact) over repeated selection of control actions over multiple performances.

However, specification of such a loss function is non-trivial. Currently, we use a simplified approach. The score following system finds the 100 ms region of the score that is most likely to encompass the performer's current position. This region is the 100 ms portion of the score position density function containing the most probability. The accompaniment system takes the center of this region as a best estimate of the current position of the vocalist. It synchronizes to this position using a set of performance rules almost identical to those described by Grubb and Dannenberg [2]. Thus the system will synchronize to the position most likely to be within 50 ms of the performer's location. The accompaniment system retains several successive position estimates for use in calculating a recent average tempo. The system adjusts its tempo and score position depending upon how closely the estimates of the performer's location and tempo correspond with its own current position and tempo.

We have incorporated the stochastic score following model as part of an automated accompaniment system. It uses a sample interval of 12 ms to represent the score position density function and responds to output from the pitch detection system previously described. This completed accompaniment system has been used to accompany both recordings of vocal performances and live singers. Recordings are the "acid" test of computer accompaniment because the recording never adjusts to compensate for the accompanist's errors as does a live performer.

6 Results

In an attempt to quantitatively assess the tracking system, we are currently examining the difference between the time at which the system estimates a particular position and the time that the vocalist was actually at that position. To do this, all position estimates made by the tracking system for a single performance are recorded with a time stamp roughly indicating the point in the audio signal last processed by the pitch detector. Also, a DAT recording of a vocal performance is played into the PC and digitized. This recording is parsed by hand to locate, as best as is possible, the onset time of each note in the score. The start of the vowel in a syllable is taken to indicate the onset. When a vowel is sustained over multiple notes, the point where pitch changes is used instead.

The start of the recorded position estimates is time aligned with the handmade transcript. Differences between the time at which the system first estimated a position within each score note and the time at which the performer was actually at that position are calculated. Graphs and statistics of this data can then be examined to assess the tracking system's accuracy, subject to the errors introduced by handmade transcripts and time alignment.

Table 1 presents summary statistics for two recorded vocal performances. These recordings are of different performers each singing a different piece. The first performance contained only one explicit tempo change, a slowing at the end of the piece, and was performed in a fairly steady manner. It contained relatively few successive note pairs with the same pitch—only 14. The system follows this recording quite well.

The second performance had two explicit tempo changes, a slowing followed by a return to tempo and a slowing at the end. It was performed in a more expressive manner with noticeable slowing at the ends of phrases. It contained 39 instances of successive note pairs having the same pitch—roughly 45% of the melodic intervals in the piece. This performance is a fairly bad case for a tracking system that relies heavily on tempo and pitch. As might be expected, the system tracks this piece less well, since several tempo alterations are unexpected and cannot be detected using pitch.

While the generated accompaniment is often reasonable, there are situations where the computer and singer are temporarily but noticeably not synchronized. These problems commonly occur in the presence of sudden, significant tempo changes that are not explicitly notated in the score. Such changes are especially troublesome if they occur while the performer is singing a sequence of notes on the same pitch. Intentional pitch changes for expressive purposes (like ornaments) are also problematic, since the actual observed pitches are given low likelihood by the observation distributions based on the score.

In instances where the vocalist intentionally and consistently modifies the performance in these ways, adjusting the event durations and the observation distributions by hand can often improve the computer's ability to track the performer. Also, since pitch and estimated tempo are not always sufficient to distinguish score position, we are currently examining extensions to the tracking model that include other relevant features from the performance. Examples include changes in amplitude indicative of note onsets and spectral features useful for speech recognition.

7 Summary and Future Work

We have presented a stochastic method of tracking a vocal performer that can be efficiently implemented as part of an automated accompaniment system. When calculating a stochastic description of the performer's current score position, this method incorporates a variety of relevant information: estimated tempo of the performer, elapsed time, and features extracted from a

Table 1: Statistics for differences between the earliest time of an estimated position within each note and the time the performer was actually at that position. Outliers were based on absolute value.

Performance	5% Outliers Removed	No. of Onsets	Minimum Time Diff.	Maximum Time Diff.	Mean Time Diff.	Standard Dev. of Time Diff.
1	No	106	-330 ms	213 ms	-29 ms	83 ms
	Yes	100	-163 ms	135 ms	-21 ms	66 ms
2	No	88	-1067 ms	337 ms	-104 ms	269 ms
	Yes	83	-433 ms	337 ms	-58 ms	192 ms

digitized waveform of the musical performance. The chief advantages of this method are the ability to characterize uncertainty of position estimates and the use of empirical or formal probability instead of heuristics when integrating diverse information relevant to tracking a vocalist.

While use of this tracking technique as part of a vocal accompaniment system appears promising, additional investigation is needed to improve performance. Analysis of digitized sound signals can yield multiple features that are relevant to tracking musical performances. It would be helpful to extend the score following model to include features other than fundamental pitch. We are also interested in comparing possible features, since a certain feature or combination of features may prove superior to others.

The density functions currently used in the model are very general. They are based upon only a few conditioning variables and a large target population of performances. It may be helpful to define more specific density functions for both the observations and the performer's motion through the score. For instance, certain tempo markings and patterns in the score will indicate specific kinds of tempo changes. Statistical descriptions of these cases would likely improve the system's tracking ability.

Finally, it could be useful to find ways of adapting the model's density functions to specific pieces and vocalists. This might involve techniques for identifying variations between the model's distribution functions and the distributions that actually result when particular vocalists perform particular pieces. Such techniques would enable localized modification of observation distributions. These modifications might gradually improve the accuracy of the tracking system when it repeatedly tracks the same performer singing the same piece.

8 Acknowledgments

We are grateful to a number of students and faculty in the department of music at Carnegie Mellon University for their help in testing the accompaniment system. Special thanks are due to Anne Elgar Kopta and Robert Fire for their assistance in recruiting vocalists, and to Geeta Bhatnagar for her performance in the video presentation. Additionally, we would like to thank Coda Music Corporation for providing us with a number of recorded vocal performances.

References

[1] Dannenberg, R. 1984. "An on-line algorithm for real-time accompaniment," Proceedings of the 1984 International Computer Music Conference, Paris, pp. 193-198.

[2] Grubb, L., Dannenberg, R. 1994. "Automating ensemble performance," Proceedings of the 1994 International Computer Music Conference, Aarhus, pp. 63-69.

[3] Inoue, W., Hashimoto, S., Ohteru, S. 1993. "A computer music system for human singing," Proceedings of the 1993 International Computer Music Conference, Tokyo, pp. 150-153.

[4] Inoue, W., Hashimoto, S., Ohteru, S. 1994b. "Adaptive karaoke system—human singing accompaniment based on speech recognition," Proceedings of the 1994 International Computer Music Conference, Aarhus, pp. 70-77.

[5] Katayose, H., Kanamori, T., Kamei, K., Nagashima, Y., Sato, K., Inokuchi, S., Simura, S. 1993. "Virtual performer," Proceedings of the 1993 International Computer Music Conference, Tokyo, pp. 138-145.

[6] Kuhn, W. 1990. "A real-time pitch recognition algorithm for music applications," Computer Music Journal, Volume 14, Number 3, pp. 60-71.

[7] Puckette, M. 1995. "Score following using the sung voice," Proceedings of the 1995 International Computer Music Conference, Banff, pp. 175-178.