

# FUZZY ANALYSIS IN PITCH CLASS DETERMINATION FOR POLYPHONIC AUDIO KEY FINDING

**Ching-Hua Chuan**

Department of Computer Science  
University of Southern California  
Integrated Media Systems Center  
Los Angeles, CA90089, USA  
chinghuc@usc.edu

**Elaine Chew**

Epstein Dep of Industrial & Systems Eng  
University of Southern California  
Integrated Media Systems Center  
Los Angeles, CA90089, USA  
echew@usc.edu

## ABSTRACT

This paper presents a fuzzy analysis technique for pitch class determination that improves the accuracy of key finding from audio information. Errors in audio key finding, typically incorrect assignments of closely related keys, commonly result from imprecise pitch class determination and biases introduced by the quality of the sound. Our technique is motivated by hypotheses on the sources of audio key finding errors, and uses fuzzy analysis to reduce the errors caused by noisy detection of lower pitches, and to refine the biased raw frequency data, in order to extract more correct pitch classes. We compare the proposed system to two others, an earlier one employing only peak detection from FFT results, and another providing direct key finding from MIDI. All three used the same key finding algorithm (Chew's Spiral Array CEG algorithm) and the same 410 classical music pieces (ranging from Baroque to Contemporary). Considering only the first 15 seconds of music in each piece, the proposed fuzzy analysis technique outperforms the peak detection method by 12.18% on average, matches the performance of direct key finding from MIDI 41.73% of the time, and achieves an overall maximum correct rate of 75.25% (compared to 80.34% for MIDI key finding).

**Keywords:** audio key finding, pitch classes, fuzzy analysis, key proximity.

## 1 MOTIVATION

Polyphonic audio key finding has gained interest in recent years, with several researchers proposing systems for extracting key from audio information [1, 2, 3, 4]. Key finding from audio typically requires several steps, including pitch class determination from audio (sometimes with pitch spelling), and key finding from pitch classes. The pitch class determination step provides pitch class information from audio signals; once the pitch class distribution has been ascertained, then a key finding algorithm can use this information to determine the key of the excerpt. In order to improve

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

the performance of existing systems, it is imperative that we should be able to segregate the sources of errors and improve on each module of the system.

In this paper, it is our goal to examine pitch class determination for key finding, to study the sources of errors, and propose a method that is tailored to reduce key finding error from audio signals. To this end, we propose a fuzzy analysis technique to refine the pitch class distribution extracted from the audio sample in order to improve the likelihood of correct key identification, and to avoid closely related keys.

Pitch class determination for audio key finding differs in several ways from pitch detection for transcription. Key finding is concerned only with determining the tonal context and not with identifying every individual note. Transcription requires the recognition of every single pitch, which includes pitch class and register information, and its duration, whereas key finding only needs pitch class information. Furthermore, contextual information such as key can often be assisted by the physical production (for example, a musician stressing structurally important pitches) and acoustic interactions (for example, the strongest harmonics tend to be pitches in the same key) of musical sound.

We have identified several sources of errors in pitch class determination for audio key finding. It is straightforward to obtain frequency information (pitch classes) from audio signals using frequency analysis methods such as the Fast Fourier Transform (FFT) [5]. Sources of errors in pitch class determination include: uneven loudness of pitches, insufficient resolution of lower frequency pitches, tuning problems, and the harmonic series effect. Although frequency analysis can identify all frequencies present in an audio segment, the louder pitches will have higher frequency spectrum values than others. Pitch perception operates on a logarithmic frequency scale, resulting in the fact that lower pitches are closer in frequency and hence harder to discriminate than higher frequencies. Pitches are often described as corresponding to discrete frequency values, which is problematic when one encounters sounds produced by instruments that are mistuned. Last but not least, each tone produced by an instrument consists not only of the fundamental frequency, but also a sequence of frequencies that are the effects of the harmonic series.

Based on experimental results, we found that audio key finding, more frequently than symbolic key finding, results in the mislabeling of pieces as being in keys

closely related to the correct one. Such errors occur because closely related keys have large overlapping pitch class sets. For example, the relative major/minor keys share exactly the same pitch classes, the only difference being their typical distributions. A slight tipping of the balance of pitch class distributions can lead to close but incorrect answers. The most common error in audio key finding is the mislabeling of a sample as being in a key that is the dominant of the actual one, for example, labeling a sample in C major as G major. Two keys related in this manner share all but one pitch class, that of the leading tone (or the seventh note in its scale). The pitch classes {C, G, E and F} feature strongly in a typical distribution of pitch classes in C major; the corresponding pitch class set for G major is {G, D, B and C}. Because both G and C are important pitches in G major, a small change in the pitch class distribution can result in G being selected as the key. In audio key finding, the problem is exacerbated by the fact that the dominant is typically the strongest harmonic of a tone other than the pitch class of the tone itself. Rather than eliminating the harmonics that help constrain the answers to closely related keys, we focus on reducing noise in the data and on the refining of the pitch class distribution to improve key recognition.

Our approach to this problem uses a fuzzy analysis technique to adjust the pitch class distribution, in light of the challenges mentioned above, so as to emphasize the correct tonal context for accurate key finding. In addition to the use of a fuzzy analysis technique to improve pitch class determination, in this paper, we also provide a methodology for evaluating the effectiveness of various pitch class determination strategies for audio key finding.

Although symbolic key finding has been studied for more than a decade and various evaluation methodologies have been employed, the evaluation for audio key finding, especially the pitch class determination part, remains obscure. There are two main problems that make the evaluation for audio key finding ill-defined. Firstly, the degree of incorrectness of closer keys (the dominant, relative, and parallel) is difficult to decide. Secondly, it is unclear how one should judge the performance of pitch class determination methods when one does not require exact pitch detection.

The difficulty of key finding varies widely across musical styles. Using the key denoted by the composer in the title, when such an answer exists, is only useful up to a degree. Consider a symphony with multiple movements: the piece stays in the main key only in the first and last movements. The second and third movements are typically composed in other keys. Even in the first movement, it is not uncommon for the piece to modulate to closely related keys, such as the dominant and relative. Furthermore, in classical music, the language gets progressively more complicated over the course of time. For example, the music in the late Romantic period is tonally much more diverse and complex than that in the Baroque period. The increasing complexity of tonal structure in pieces poses another challenge for evaluation of audio key finding. However,

the eradication of closer key errors in single-key examples must to be solved before audio key finding can advance further to account for modulations.

In this paper, we provide an evaluation methodology for pitch class determination in audio key finding by comparing the key assignment results for symbolic and audio music using Chew's Spiral Array Center of Effect Generator (CEG) key finding algorithm (see [6, 7]). The test sets for audio are rendered from MIDI so that the audio key finding will use exactly the same test set as that in MIDI. By using the same key finding algorithm and test sets, we can compare the results of MIDI and audio key finding to evaluate the performance of the proposed pitch class determination technique.

We used our system to test 410 pieces of classical music across a wide spectrum of time periods ranging from Baroque to Contemporary. The selection consists of a wide range of tonal music so that the aggregate results will be as unbiased as possible. Detailed analysis of results for each musical period is also provided.

The remainder of the paper is organized as follows: Section 2 provides a literature review of work in audio key finding, Section 3 describes the overall system diagram and introduces each part of the system, including the new fuzzy analysis technique, the experimental design and results are presented in Section 5, and conclusions and future work follow in Section 6.

## 2 BACKGROUND

The approach in this paper differs from previous efforts in three ways, the obvious two being the fuzzy analysis technique for pitch class determination and the use of the Spiral Array CEG algorithm for key finding. The third distinguishing feature is the systematic examination of the parts of a key finding system, and careful evaluation to isolate and measure the improvements provided by changes to a particular component.

Audio key finding systems require two basic components: one comprising of some pitch class determination method such as the generating of pitch class distributions, and another consisting of some key finding algorithm that determines the key given a pitch class distribution. In this paper, we aim to improve the pitch class determination method by using a fuzzy analysis technique and to systematically evaluate its effectiveness by comparing it to symbolic key finding. Most research in audio key finding fails to discriminate among the sources of errors, reporting only the overall system's key finding results [1, 2, 4].

In Gómez [1] and in Gómez and Herrera [2], the authors detected pitches using thrice the standard resolution of the pitch frequency spectrum of the FFT method, and distributed the frequency values among the adjacent frequency bins using a weighting function to reduce boundary errors. They generated a Harmonic Pitch Class Profile as input to Krumhansl and Schmuckler's (K-S) key finding method [8]. Their template pitch class profile gives the dominant a higher weight than the tonic, a counterintuitive assignment.

They reported an overall correct rate of 66.1% when testing on 833 pieces of classical and jazz pieces [2].

Pauws [4] incorporated rules for avoiding noise and emphasizing pitch loudness in his pitch class determination method, and applied the K-S method to generate the key. Pauws used 237 classical piano sonatas as the test set and his method resulted in a correct rate of 59.1% within 5 seconds, and achieved a maximum correct rate of 66.2% within 15 seconds.

It is unclear if the errors reported by the authors of these two audio key finding systems are due in larger part to their pitch class determination techniques, or to the key finding algorithm. The two systems differ primarily in the pitch class determination step, in the way in which the systems generate the pitch class profile for the standard FFT results. Both used the K-S probe tone method [8] for key finding, with Gómez using a modified template profile.

In 1996, Izmirli and Bilgen presented a model that analyses tonal context as a continuous function [3]. They used a constant Q transform to generate pitch classes and proposed a leaky integrator based on the K-S model to determine the tonal center. However, in their study, only two music excerpts are evaluated, a less than representative sample size.

There exists only a limited number of models for key finding. In 1986, Krumhansl and Schmuckler (K-S model) [8] proposed the probe tone profile method that matches pitch duration profiles to template pitch class profiles for major and minor keys, acquired from user ratings of probe tone experiments. The key is determined as the one with the highest correlation value. In 1999, Temperley improved upon the K-S method by modifying the template pitch class profiles through musical reasoning [9]. Temperley modified the profiles to emphasize the differences between diatonic and chromatic scales, and also adjusted the weights of the fourth and seventh pitches so as to differentiate the keys with highly similar pitch class sets.

In this paper, we employ the Spiral Array CEG algorithm proposed by Chew [6, 7] to determine key for MIDI and audio. The Spiral Array Model is a 3-dimensional model that represents pitches, intervals, chords and keys in the same space for easy comparison. On the Spiral Array, pitches are represented as points on a helix, and adjacent pitches are related by intervals of perfect fifths, while vertical neighbors are related by major thirds. In the CEG algorithm, key selection is performed by summarizing musical information as a spatial point in the interior of the spiral and by conducting a nearest neighbor search in the Spiral Array space. Although the K-S model is one of the most widely used key finding methods, the Spiral Array CEG model has been demonstrated to achieve better key finding results using symbolic data sets [6, 7]. We implemented both the Spiral Array CEG method and the K-S model in an earlier audio key finding system that employed simply peak detection from FFT [10]. We observed that the CEG method again consistently outperformed the K-S method with few exceptions. In this paper, we will use the Spiral Array CEG method as

a constant among the three systems we test: key finding from MIDI, key finding from audio using only peak detection, and key finding from audio using peak detection and fuzzy analysis.

### 3 SYSTEM DESCRIPTION

In our proposed audio key finding system, we employ a fuzzy analysis technique, with adaptive weights and periodic cleanup, to generate a pitch class distribution that reduces closer-key errors in key finding. Figure 1 shows a diagram of the system for polyphonic audio key finding.

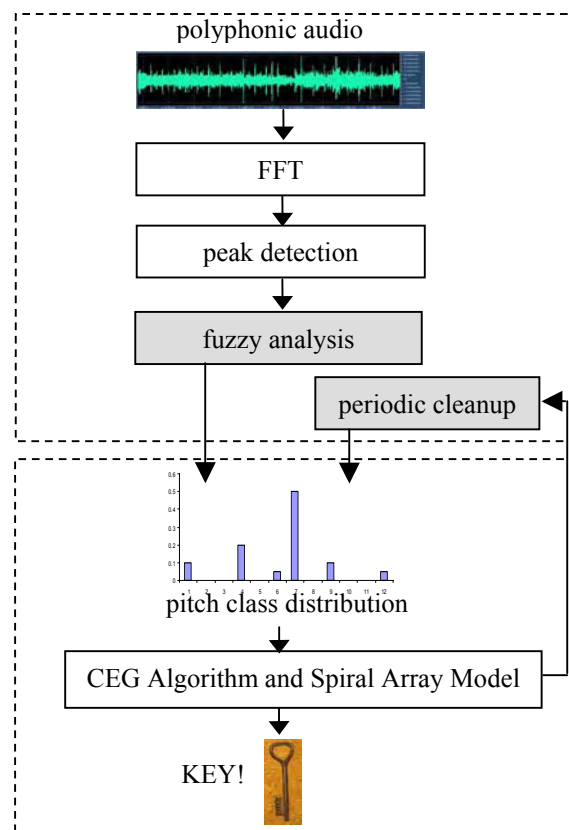


Figure 1. Graph of audio key finding system.

The system consists of two major parts. The first generates a distribution of weights for the twelve pitch classes from audio signals, as shown in the upper dashed box in Figure 1. We use the FFT to extract the frequency information, and employ the peak detection method described in [10] and in Section 3.1. We apply a fuzzy analysis technique (detailed in Section 3.2) and a periodic cleanup procedure (explained in Section 3.3) to generate refined weight distributions for the key finding algorithm in order to increase the likelihood of obtaining the correct key. This fuzzy analysis method will be described in detail in Section 3.

The second part of the system contains the key finding algorithm. This module consists of pitch spelling and key finding. To represent pitch class information for comparison to key representations, we use Chew's Spiral Array Model [6, 7]. The pitch spelling method for mapping numeric pitch classes to

letter name pitch class representations on the pitch spiral is described in [11] and [12]. Finally, we employ the CEG algorithm [6, 7, 13] to determine the key.

In Section 4, we will compare three systems for key finding. The first, key finding from MIDI is contained within the lower dashed box in Figure 1: this system takes MIDI files as input, generates pitch classes, and uses the Spiral Array CEG algorithm to determine the key. The second system performs audio key finding using the FFT and a peak detection method [10] in the pitch class determination phase (henceforth referred to as the audio key finding with peak detection); this system is represented by all modules in the diagram except for the gray boxes in Figure 1. The third is the audio key finding system with the fuzzy analysis technique and periodic cleanup procedure. The entire system is shown in Figure 1.

In the following sections we describe the methods designed to reduce noise and refine the pitch class distribution in the pitch class determination phase.

### 3.1 Pitch frequency detection from audio signal

We use standard FFT with the peak detection method described in [10] to extract the corresponding frequency magnitude for each pitch. The peak detection method selects the local maximum within the frequency range pre-defined for each pitch. In [10], we summed these maximum values for all pitches in a class to get the pitch class distribution. Instead of directly using the local maximum values to generate the pitch class distributions, here, we apply a fuzzy analysis technique (described in the next section) to refine the local maxima so as to obtain more accurate pitch class distributions that avoid closer-key errors.

### 3.2 Fuzzy analysis with adaptive level weights

We use a fuzzy analysis technique to clarify pitch information from the frequency spectrum. Pitch class detection from audio signal is inherently noisy for the reasons outlined in Section 1 – uneven loudness of pitches, harmonic series effect, and insufficient resolution in lower frequencies, increase the difficulty of recognizing polyphonic audio pitch – resulting in the incorrect detection of closely related keys, such as the dominant, relative and parallel keys. The errors also accumulate, thus worsening the performance of key finding systems over time.

Our method consists of three steps. The first two aims to clarify information in the lower frequencies. The first step, detailed in Section 3.2.1, uses knowledge of the overtone series to clarify membership in the lower ranges. The second step, described in Section 3.2.2, scales the FFT results in each pre-defined range by the density of the signal in that range so as to properly acknowledge the presence of important pitches in that frequency range. This second step, called adaptive level weighting, is particularly effective in the clarifying of low pitches in late romantic music, especially music that begins almost exclusively in the low frequency ranges. After the pitch values have been folded into pitch class

values, we employ the third and final step to refine the pitch class distribution. The third step, outlined in Section 3.2.3, sets all normalized pitch class values 0.2 and below to zero, and all values 0.8 and above to one.

The reason that we use fuzzy analysis is that one can view the results of an FFT analysis as a fuzzy description, not a calculated probability, of the likelihood that a pitch is played. The fuzzy analysis technique provides several advantages: (1) it generates more accurate weight distributions for pitch classes to determine the correct key, instead of closely related ones; and, (2) By employing adaptive level weights, the technique is robust against the impact of different musical arrangements (the varying of registers and instrumentation) and styles.

#### 3.2.1 Clarifying low frequencies

In Step 1, we use the overtone series as the basis for fuzzy analysis to clarify pitches of frequencies below 261 Hz (the pitch C<sub>4</sub>). Pitch frequencies are defined on a logarithmic scale; thus, the frequencies of lower pitches are more closely spaced than higher ones. The mapping of lower frequencies to their pitches, defined by discrete frequency ranges, is particularly noisy and error prone. In contrast, assignment of higher-frequencies to pitches is more accurate because of the relatively wider frequency ranges. Therefore, we use the presence of the first overtone to determine and to refine the weights for lower pitches.

We use the idea of the membership value in fuzzy logic to represent the likelihood that a pitch has been sounded. The membership values are based on the FFT results after peak detection. We set the highest peak, P<sub>max</sub>, the pitch membership value of the largest FFT result, to one, which one can interpret as assuming that this pitch is definitely sounded. We get the membership values for all other pitches by dividing their peak values by P<sub>max</sub>. We set the values less than 0.1 to zero to eliminate some noise. The value 0.1 was chosen after performing some preliminary tests. Assume that P<sub>ij</sub> represents the pitch of class *j* at register *i*, for example, middle C (C<sub>4</sub>) is P<sub>4,0</sub>. Let FFT(P<sub>ij</sub>) be the local peak for pitch P<sub>ij</sub> after the FFT. Then, the membership value for P<sub>ij</sub> is defined as:

$$mem(P_{i,j}) = FFT(P_{i,j}) / P_{max}, \quad (1)$$

where *i* = 2, 3, 4, 5, 6, and *j* = 1...12, which allows for pitches ranging from C<sub>2</sub> (65 Hz) to B<sub>6</sub> (2000 Hz).

By examining the membership values of the pitch an octave above, the pitch one half-step above and its first overtone, we can remove the common errors caused by insufficient frequency resolution and discrete frequency definition in lower pitches. The method nullifies the membership value of the lower pitch for which the pitch one half step above has a membership value higher than its own, or the pitch one octave higher or the pitch one half step and one octave higher has a membership value higher than its own. A reason for this nullifying step is that if the pitch one half step above has the higher membership value, then it is likely that the present pitch is incorrectly detected. Another reason is that pitch

membership values are more accurate in the higher registers than in lower ones. Hence, if the membership value of the pitch an octave above is higher, then this higher value already accounts for that pitch class in the final distribution. If the membership value of the pitch a half step plus an octave above is higher, then it is likely that the current one was not sounded.

Mathematically, we first define the *membership negation value* for lower pitches, a quantity that represents the fuzzy likelihood that a pitch is not sounded. The membership negation value is the maximum of the membership values of the pitch one half step above ( $P_{i,j+1}$ ), and the first overtones of the pitch itself ( $P_{i+1,j}$ ) and that of the pitch one half step above ( $P_{i+1,j+1}$ ):

$$\sim mem(P_{i,j}) = \max\{mem(P_{i,j+1}), mem(P_{i+1,j}), mem(P_{i+1,j+1})\}, \quad (2)$$

where  $i = 2, 3$  and  $j = 1 \dots 12$ , because we consider only the lower frequency pitches, pitches below  $C_4$ .

Next, we set the membership values of lower-frequency pitches to zero if its membership negation value is larger than its membership value:

$$mem(P_{i,j}) = \begin{cases} 0, & \text{if } \sim mem(P_{i,j}) > mem(P_{i,j}) \\ mem(P_{i,j}), & \text{if } \sim mem(P_{i,j}) \leq mem(P_{i,j}) \end{cases}, \quad (3)$$

where  $i = 2, 3$  and  $j = 1 \dots 12$ .

### 3.2.2 Adaptive level weighting

The fuzzy analysis technique described in the previous section prioritizes the membership values of higher frequency pitches. This becomes problematic in key evaluation of pieces containing large segments of music with only lower frequency pitches. The adaptive level weighting scheme described here scales the FFT results in each pre-defined range by the density of the signal in that range so as to better detect the presence of important pitches in that frequency range.

The adaptive level weight for a given range, a scaling factor, is the relative density of signal in that range. For example, the adaptive level weight for register  $i$  (which includes pitches  $C_i$  through  $B_i$ ),  $Lw_i$ , is defined as:

$$Lw_i = \frac{\sum_{j=1}^{12} FFT(P_{i,j})}{\sum_{k=2}^6 \sum_{j=1}^{12} FFT(P_{k,j})}, \quad (4)$$

Finally, we generate the weight for each pitch class,  $mem(C_j)$ , by summing the membership values of that pitch across all registers, multiplied by the corresponding adaptive level weight:

$$mem(C_j) = \sum_{i=2}^6 Lw_i * mem(P_{i,j}), \text{ where } j = 1 \dots 12. \quad (5)$$

### 3.2.3 Flatten high and low values

To reduce minor differences in the membership values of important pitch classes and to eliminate low-level noise, we introduce the final step described in this section. We set the pitch class membership values equal to one if they are larger than 0.8, and equal to zero if

they are less than 0.2. The flat output for higher membership values prevents louder pitches from dominating the weight distribution. Last but not least, we normalize the membership values for all pitch classes by scaling them to sum to one.

## 3.3 Periodic cleanup

Based on our observations, errors tend to accumulate over time. To counter this effect, we implemented a periodic cleanup procedure that takes place every 2.5 seconds. In this cleanup step, we sort the pitch classes in ascending order and isolate the four pitches with the smallest membership values. We set the two smallest values to zero, a reasonable choice since most scales consist of only seven pitch classes. For the pitch classes with the third and fourth smallest membership values, we consult the current key assigned by the CEG algorithm; if the pitch class does not belong to the key, we set the membership value to zero as well.

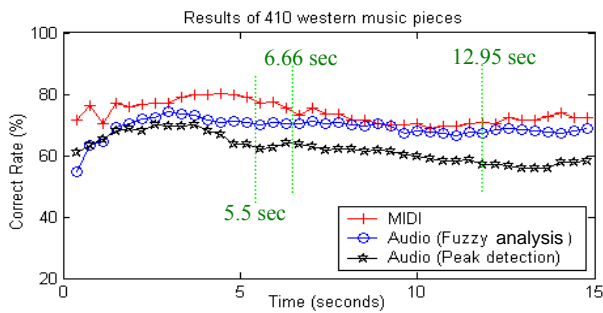
## 4 EXPERIMENTS AND RESULTS

To evaluate the fuzzy analysis technique, we choose excerpts from 410 classical music pieces by various composers across different time and stylistic periods, ranging from Baroque to Contemporary. Most the pieces are concertos, preludes, and symphonies, comprising of polyphonic sounds from a variety of instruments. The key of each piece is stated explicitly by the composer in the title. We use only the first fifteen seconds of the first movement, so that the test samples are highly likely to remain in the stated key for the entire duration of the sample.

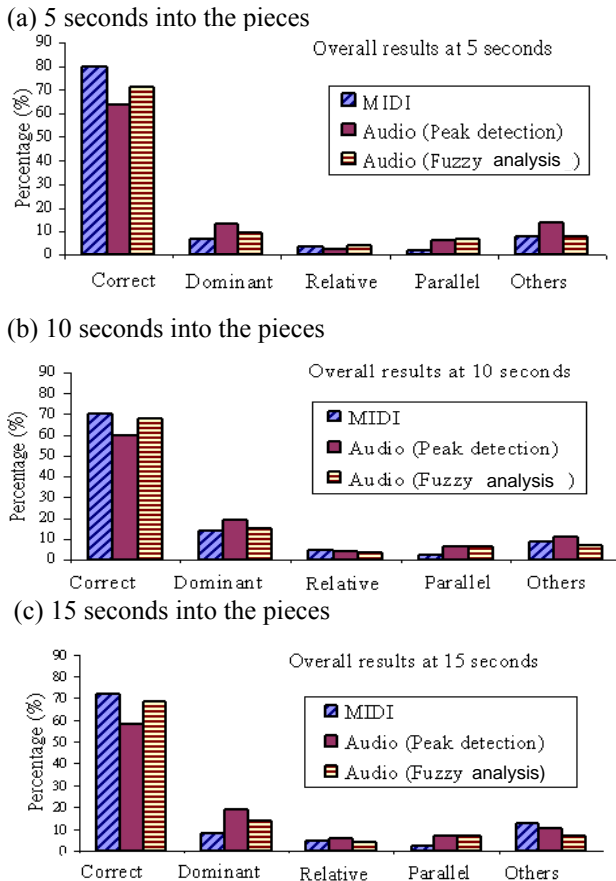
In order to facilitate the comparison of audio key finding from symbolic and audio data, we started with MIDI samples from [www.classicalarchives.com](http://www.classicalarchives.com), and used the Winamp software with a sampling rate of 44.1 kHz to render MIDI files to audio (wave format). We tested three different systems on the same pieces. The first system applied the CEG algorithm to MIDI files, the second applied the CEG algorithm to pitch class distributions generated by peak detection, and the third applied the CEG algorithm to pitch class distributions generated by fuzzy analysis. Each system returned a key answer every 0.37 seconds and the answers are classified into five categories: correct, dominant, relative, parallel, and others.

### 4.1 Overall results

The correct rates of the three systems over time are shown in Figure 2. For the 410 classical music pieces, the correct rate using fuzzy analysis is consistently higher than that for the peak detection method, except for the first 0.37 seconds. The difference exceeds 10% from 5.55 seconds onwards. From 6.66 to 12.95 seconds, the results of the audio key finding system with fuzzy analysis perform almost as well as that for MIDI key finding.



**Figure 2.** Comparison of overall key finding results.



**Figure 3.** Detailed analysis of overall results.

The detailed analyses of the results at 5, 10 and 15 seconds are shown in Figures 3(a), (b), and (c) respectively. In Figure 3, the results are classified into five categories. Notice that most of the closer-key errors for audio key finding were due to the mislabelling of the pieces as being in the dominant key (a perfect fifth above the correct one). The fuzzy analysis technique improves the correct rate by reducing the closer-key answers in the Dominant and the Others categories. The audio key finding methods (both peak detection and fuzzy analysis) suffer more from parallel key errors, while the symbolic (MIDI) key finding suffers more from relative key errors. These results are probably due to the fact that relative keys share the same pitch classes as the correct one, while the audio dominant and parallel key errors most likely result from incorrect weight distributions in the pitch classes.

The other important observation is that symbolic key finding suffers most from errors in the Others category 15 seconds into the pieces, as shown in Figure 3(c). One explanation for this could be that symbolic key finding is distracted by extraneous information such as accidentals, while the amplitude and frequency characteristics of musical audio signals constrain audio key finding results to mostly the closer keys.

The resulting maximum correct percentage, average correct percentage, and median correct percentage for key identification by the three systems are summarized in Table 1. Notice that the fuzzy analysis technique significantly improves the peak detection results, especially in terms of the average correct percentage and median correct percentage.

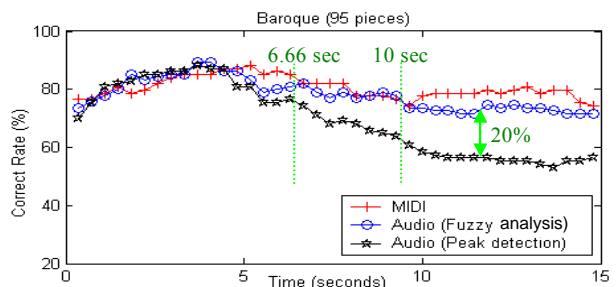
Table 1. Summary of overall results.

	MIDI	Audio (peak detection)	Audio (fuzzy analysis)
Max correct percentage (%)	80.34	70.17	75.25
Average correct percentage (%)	73.91	62.23	69.81
Median correct percentage (%)	72.97	61.98	70.22

#### 4.2 Results sorted by stylistic period

We classify our test data according to the stylistic periods defined by [www.classicalarchives.com](http://www.classicalarchives.com): Baroque (Bach and Vivaldi), Classical (Haydn and Mozart), Late Classical and Early Romantic (Beethoven and Schubert), Romantic (Chopin, Mendelssohn, and Schumann), Late Romantic (Brahms and Tchaikovsky), and Contemporary (Copland, Gershwin, and Shostakovich).

The key finding results for 95 pieces by Bach and Vivaldi (concertos) are shown in Figure 4. The audio key finding systems perform as well as, sometimes even superceding, the MIDI key finding results in the first 5 seconds. The results for the peak detection method drop after 6.66 seconds. In contrast, the correct rate of the fuzzy analysis technique remains comparable to that for MIDI. The correct rate for the fuzzy analysis technique is more than 20% higher than the peak detection method from 10 seconds to 15 seconds.

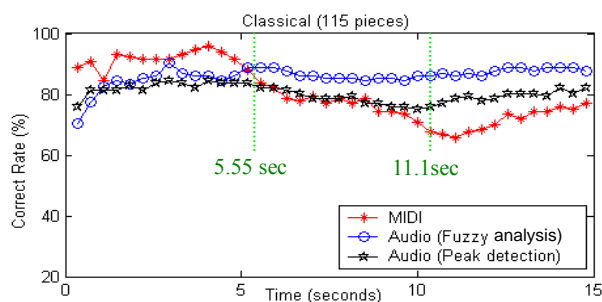


**Figure 4.** Key finding results for 95 Baroque pieces.

The key finding results for 115 pieces by Haydn and Mozart (symphonies) are shown in Figure 5. The test samples from the classical period are the only cases where the audio key finding systems outperform that for

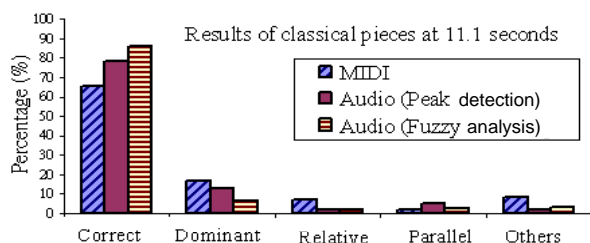


MIDI. Note that the system with fuzzy analysis has a higher correct rate than that with peak detection from 5.55 to 15 seconds.



**Figure 5.** Key finding results for 115 classical pieces by Haydn and Mozart.

At 11.1 seconds, we observe the largest difference between MIDI and audio key finding. Figure 6 presents the detailed breakdown of the key finding results at 11.1 seconds. The figure shows that most of the MIDI errors are due to assignments to the dominant key. It is difficult to judge from the summary statistics whether some of the pieces actually change to their dominant keys at this time, a likely scenario. However, the symbolic key finding system produces more errors in the Others category than the audio key finding systems.



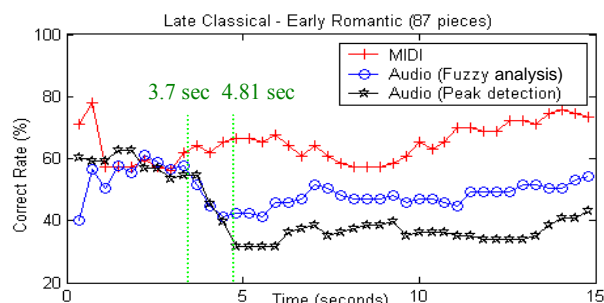
**Figure 6.** Detailed analysis of key finding results for 115 classical pieces at 11.1 seconds.

The results of Late Classical-Early Romantic, Romantic, and Late Romantic are presented in Figure 7(a), (b), and (c) respectively. Compared to the Baroque and Classical periods, the correct rates are significantly diminished for examples from each of these later periods. The shape of the correct percentage line reflects the less structured music style. For example, in Romantic period, the results for all systems start with lower correct rates and gradually increase over time. In the Late Romantic period, the results also start with a lower correct rate, then increase significantly within 2.59 seconds, but drop again towards 15 seconds. The results imply that in the Romantic period, the music may start with a key other than the one stated, while in the Late Romantic period, the music changes quickly to other keys.

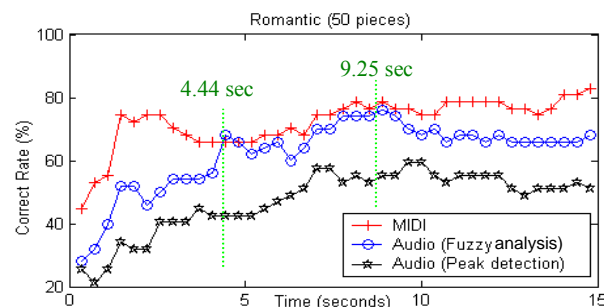
Figure 8 shows the key finding results for 29 pieces by Copland, Gershwin and Shostakovich, twentieth century classical composers. Observe that the results in Figures 7(a), (b), (c), and 8 show higher correct rates for the fuzzy technique than the peak detection method. In each case, the fuzzy technique resulted in correct rates

closer or equal to the MIDI results. In Figure 7(a), the audio results decrease at 3.7 seconds, but fuzzy analysis has a beneficial effect on audio key finding between 4.81 and 15 seconds. For music from the Romantic period (Figure 7(b)), the fuzzy analysis results are comparable to that for symbolic key finding between 4.44 and 9.25 seconds. For music from the Late Romantic period (Figure 7(c)), MIDI and audio key finding perform similarly, with results that are difficult to verify objectively due to the tonal complexity of the pieces. In Figure 8, the fuzzy analysis technique obtains better results than the peak detection method from 2.22 seconds to 8.51 seconds, while the MIDI results are consistently better. This could be due in part to pitch spelling errors, as the pitch spelling technique gives priority to pitches in the same key (as does the periodic cleanup procedure).

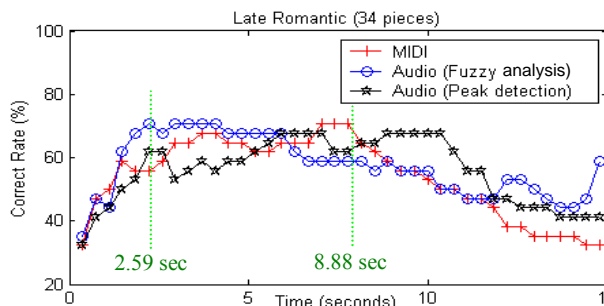
(a) Late Classical and Early Romantic: 87 pieces by Beethoven and Schubert



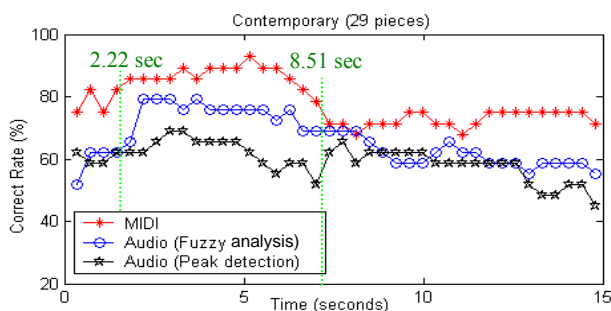
(b) Romantic: 50 pieces by Chopin, Mendelssohn, and Schumann



(c) Late Romantic: 34 pieces by Brahms and Tchaikovsky



**Figure 7.** Key finding results for late classical and romantic works.



**Figure 8.** Key finding results for 29 Contemporary pieces by Copland, Gershwin and Shostakovich.

## 5 CONCLUSIONS

We have presented a fuzzy analysis technique for pitch class determination to improve the accuracy of key finding for polyphonic audio. We evaluate the technique by comparing the results of symbolic (MIDI) key finding, audio key finding with peak detection, and audio key finding with fuzzy analysis. We showed that the fuzzy analysis technique was superior to a simple peak detection policy, increasing the percentage of correct key identifications by 12.18% on average. In fact, the percentage correct for the fuzzy analysis technique matched that of symbolic key finding 41.73% of the time.

## REFERENCES

- [1] Gómez, E., "Tonal Description of Polyphonic Audio for Music Content Processing", *INFORMS Journal on Computing*, to appear.
- [2] Gómez, E. Herrera, P., "Estimating The Tonality Of Polyphonic Audio Files: Cognitive Versus Machine Learning Modelling Strategies", *ISMIR 2004 – 5<sup>th</sup> International Conference on Music Information Retrieval*, 2004.
- [3] Izmirli, O. and Bilgen, S., "A Model for Tonal Context Time Course Calculation from Acoustical Input," *Journal of New Music Research*, Vol. 25(3), 1996.
- [4] Pauws, S., "Musical Key Extraction from Audio", *ISMIR 2004 – 5<sup>th</sup> International Conference on Music Information Retrieval*, 2004.
- [5] Mitra, Sanjit K., *Digital Signal Processing: A Computer Based Approach*, 2<sup>nd</sup> Edition. McGraw-Hill, 2001.
- [6] Chew, E. *Towards a Mathematical Model of Tonality*. Doctoral dissertation, Department of Operations Research, Massachusetts Institute of Technology, Cambridge, MA, 2000.
- [7] Chew, E., "Modeling Tonality: Applications to Music Cognition", *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, Edinburgh, Scotland, 2001.

- [8] Krumhansl, C.L., *Quantifying Tonal Hierarchies and Key Distances*. *Cognitive Foundations of Musical Pitch*, chapter 2, 16-49, 1990.
- [9] Temperley, D., "What's Key for Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered," *Music Perception*, 17(1), 65-100, 1999.
- [10] Chuan, C. H. and Chew, E., "Polyphonic Audio Key Finding Using the Spiral Array CEG Algorithm", *Proceedings of the IEEE International Conference on Multimedia and Expo*, Amsterdam, The Netherlands, 2005.
- [11] Chew, E. and Chen, Y. C., "Mapping MIDI to the Spiral Array: Disambiguating Pitch Spellings", H. K. Bhargava and Nong Ye (Eds.), *Computational Modeling and Problem Solving in the Networked World*, Kluwer, pp.259-275. *Proceedings of the 8th INFORMS Computer Society Conference, ICS2003*, Chandler, AZ, Jan 8-10, 2003.
- [12] Chew, E. and Chen, Y. C., *Real-Time Pitch Spelling Using the Spiral Array*. *Computer Music Journal*. 29:2, Summer 2005.
- [13] Chew, E. "Slicing it all ways: mathematical models for tonal induction, approximation and segmentation using the Spiral Array", *INFORMS Journal on Computing*, to appear.