Multiple Fundamental Frequency Estimation by Modeling Spectral Peaks and Non-Peak Regions

Zhiyao Duan, *Student Member, IEEE*, Bryan Pardo, *Member, IEEE*, and Changshui Zhang, *Member, IEEE*

***Abstract*—This paper presents a maximum-likelihood approach to multiple fundamental frequency (F0) estimation for a mixture of harmonic sound sources, where the power spectrum of a time frame is the observation and the F0s are the parameters to be estimated. When defining the likelihood model, the proposed method models both spectral peaks and non-peak regions (frequencies further than a musical quarter tone from all observed peaks). It is shown that the peak likelihood and the non-peak region likelihood act as a complementary pair. The former helps find F0s that have harmonics that explain peaks, while the latter helps avoid F0s that have harmonics in non-peak regions. Parameters of these models are learned from monophonic and polyphonic training data. This paper proposes an iterative greedy search strategy to estimate F0s one by one, to avoid the combinatorial problem of concurrent F0 estimation. It also proposes a polyphony estimation method to terminate the iterative process. Finally, this paper proposes a postprocessing method to refine polyphony and F0 estimates using neighboring frames. This paper also analyzes the relative contributions of different components of the proposed method. It is shown that the refinement component eliminates many inconsistent estimation errors. Evaluations are done on ten recorded four-part J. S. Bach chorales. Results show that the proposed method shows superior F0 estimation and polyphony estimation compared to two state-of-the-art algorithms.**

***Index Terms*—Fundamental frequency, maximum likelihood, pitch estimation, spectral peaks.**

I. Introduction

MULTIPLE fundamental frequency (F0) estimation in polyphonic music signals, including estimating the number of concurrent sounds (polyphony), is of great interest to researchers working in music audio and is useful for many applications, including automatic music transcription [1], source separation [2], and score following [3]. The task, however, remains challenging and existing methods do not match human ability in either accuracy or flexibility.

All those who develop multiple F0 estimation systems must make certain design choices. The first of these is how to preprocess the audio data and represent it. Some researchers do not employ any preprocessing of the signal and represent it with the full time domain signal or frequency spectrum. In this category, discriminative model-based [1], generative model-based [4], [5], graphical model-based [6], spectrum modeling-based [7]–[11], and genetic algorithm-based [12] methods have been proposed.

Because of the high dimensionality of the original signal, researchers often preprocess the audio with some method to retain salient information, while abstracting away irrelevant details. One popular data reduction technique has been to use an auditory model to preprocess the audio. Meddis and Mard [13] proposed a unitary model of pitch perception for single F0 estimation. Tolonen and Karjalainen [14] simplified this model and applied it to multiple F0 estimation of musical sounds. de Cheveigné and Kawahara [15] integrated the auditory model and used a temporal cancellation method for F0 estimation. Klapuri [16], [17] used auditory filterbanks as a front end, and estimated F0s in an iterative spectral subtraction fashion. It was reported that [17] achieves the best performance among methods in this category.

Another more compact data reduction technique is to reduce the full signal (complex spectrum) to observed power spectral peaks [18]–[24]. The rationale is that peaks are very important in terms of human perception. For example, resynthesizing a harmonic sound using only peaks causes relatively little perceived distortion [25]. In addition, peaks contain important information for pitch estimation because, for harmonic sounds, they typically appear near integer multiples of the fundamental frequency. Finally, this representation makes it easy to mathematically model the signal and F0 estimation process. Given these observations, we believe this representation can be used to achieve good results.

Section I-A reviews the methods that focus on estimating F0s from detected peaks, which are closely related to our proposed method.

A. Related Work

Goldstein [18] proposed a method of probabilistic modeling of peak frequencies for single F0 estimation. Given an F0, energy is assumed to be present around integer multiples of the F0 (the *harmonics*). The likelihood of each spectral peak, given the F0, is modeled with a Gaussian distribution of the frequency deviation from the corresponding harmonic. The best F0 is presumed to be the one that maximizes the likelihood of generating

Manuscript received May 26, 2009; revised January 07, 2010. Date of publication February 02, 2010; date of current version September 08, 2010. This work was supported in part by the U.S. National Science Foundation under Grant IIS-0643752, in part by a China 973 Program (2009CB320602), and in part by the China National Science Foundation under Grant 60721003. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sylvain Marchand.

Z. Duan and B. Pardo are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: zhiyaoduan2012@u.northwestern.edu; pardo@cs.northwestern.edu).

C. Zhang is with the State Key Lab of Intelligent Technologies and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation–Tsinghua University, Beijing 100084, China (e-mail: zcs@mail.tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2010.2042119

the set of peak frequencies in the observed data. This model does not take into account observed peak amplitudes.

Thornburg and Leistikow [20] furthered Goldstein's idea of probabilistic modeling of spectral peaks. Given an assumed F0 and the amplitude of its first harmonic, a template of ideal harmonics with exponentially decaying amplitudes is formed. Then, each ideal harmonic is uniquely associated with at most one observed spectral peak. This divides peaks into two groups: *normal peaks* (peaks associated with some harmonics) and *spurious peaks* (peaks not associated with harmonics). The probability of every possible peak-harmonic association is modeled. All possible associations are marginalized to get the total likelihood, given an F0. They account for spurious peaks in this formulation to improve robustness. Leistikow *et al.* [21] extended the above work to the polyphonic scenario. The modeling and estimating methods remain the same, except that when forming the ideal harmonic template, overlapping harmonics are merged as one harmonic.

The methods in [20] and [21] achieve good results. However, the computational cost can be heavy, since the association between harmonics and peaks is subject to a combinatorial explosion problem. They deal with this by approximating the exact enumeration with a Markov Chain Monte Carlo (MCMC) algorithm. Furthermore, both papers assume known-good values for a number of important parameters (the decay rate of harmonic amplitudes, the standard deviation of Gaussian models, the parameters in the probability of the association, etc.). The approach in [21] also assumes the polyphony of the signal is known. This can be problematic if the polyphony is unknown or changes as time goes by.

The above methods output the F0 estimate(s) whose predicted harmonics best explain spectral peaks. This, however, may tend to overfit the peaks. An F0 estimate which is one octave lower of the true F0 may explain the peaks well, but many of its odd harmonics may not find peaks to explain.

Maher and Beauchamp [19] noticed this problem and proposed a method for single F0 estimation for quasi-harmonic signals. Under the assumption that the measured partials (spectral peaks) have a one-to-one correspondence with the harmonics of the true F0, a two-way mismatch (TWM) between measured partials and predicted harmonics of a F0 hypothesis is calculated. The F0 hypothesis with the smallest mismatch between predicted and measured partials is selected.

Recently, this idea was also adopted by Emiya *et al.* [11] in multiple F0 estimation for polyphonic piano signals. In [11], each spectrum is decomposed into the sinusoidal part and the noise part. A weighted maximum-likelihood model combines these two parts, with the objective of simultaneously whitening the sinusoidal sub-spectrum and the noise sub-spectrum.

B. Advances of Proposed Method

In our work, we address the multiple F0 estimation problem in a Maximum-Likelihood fashion, similar to [18], [20], and [21], adopting the idea in [11] and [19] and building on previous results in [22]. We model the observed power spectrum as a set of peaks and the non-peak region. We define the *peak region* as the set of all frequencies within d of an observed peak. The *non-peak region* is defined as the complement of the peak region (see Section III for detailed definitions). We then define a likelihood on both the peak region and the non-peak region, and the total likelihood function as their product. The peak region likelihood helps find F0s that have harmonics that explain peaks, while the non-peak region likelihood helps avoid F0s that have harmonics in the non-peak region. They act as a complementary pair. We adopt an iterative way to estimate F0s one by one to avoid the combinatorial problem of concurrent F0 estimation.

TABLE I
Proposed Multi-F0 Estimation Algorithm

```
For each frame of audio
    find peak frequencies and amplitudes with [26]
    $\mathcal{C}$ = a finite set of frequencies within $d$ of peak freqs
    $\boldsymbol{\theta} = \emptyset$
    For $N$ = 1 to MaxPolyphony
        For each $F_0$ in $\mathcal{C}$
            Evaluate Eq. (2) on $\boldsymbol{\theta} \bigcup \{F_0\}$          (Section III)
        Add to $\boldsymbol{\theta}$ the $F_0$ that maximized Eq. (2)
    Estimate actual polyphony $N$ with Eq. (18)          (Section IV)
    Return the first $N$ estimates in $\boldsymbol{\theta} = \{F_0^1, \cdots, F_0^N\}$
For each frame of the audio
    Refine F0 estimates using neighboring frames          (Section V)
```

Our method is an advance over related work in several ways. First, our likelihood model avoids the issue of finding the correct associations between every possible harmonic of a set of F0s and each observed peak as in [20] and [21]. Instead, each peak is considered independently. The independence assumption is reasonable, since a stronger assumption that all the spectral bins are conditionally independent, given F0s, is commonly used in literature [4]. Because of this, the likelihood computational cost is reduced from $O(2^K)$ to $O(K^2)$, where K is the number of spectral peaks. Therefore, our method can be evaluated on a relatively large data set of real music recordings, while [18], [20], [21] are all tested on a small number of samples.

Second, we adopt a data-driven approach and parameters are all learned from monophonic and polyphonic training data (summarized in Table II), while model parameters are all manually specified in [11], [18], [20], and [21].

Third, we use a simple polyphony estimation method that shows superior performance compared to an existing method [17]. Recall that the most closely related method [21] to our system requires the polyphony of the audio as an input.

Finally, our method uses a postprocessing technique to refine F0 estimates in each frame using neighboring frames, while related methods do not use local context information. Experimental results show our use of local context greatly reduces errors.

The remainder of this paper is arranged as follows. Section II gives an overview of the system. Section III presents the model to estimate F0s when the polyphony is given. Section IV describes how to estimate the polyphony. Section V describes the postprocessing technique. Section VI presents an analysis of computational complexity. Experiments are presented in Section VII, and the paper is concluded in Section VIII.

TABLE II
PARAMETERS LEARNED FROM TRAINING DATA. THE FIRST FOUR PROBABILITIES ARE LEARNED FROM THE POLYPHONIC TRAINING DATA. THE LAST ONE IS LEARNED FROM THE MONOPHONIC TRAINING DATA

$P(s_k)$	Prob. a peak k is normal or spurious
$p(f_k, a_k \mid s_k = 1)$	Prob. a spurious peak has frequency f_k and amplitude a_k
$p(a_k \mid f_k, h_k)$	Prob. a normal peak has amplitude a_k, given its frequency f_k and it is harmonic h_k of an F0
$p(d_k)$	Prob. a normal peak deviates from its corresponding ideal harmonic frequency by d_k
$P(e_h \mid F_0)$	Prob. the h-th harmonic of F_0 is detected

II. System Overview

Table I shows the overview of our approach. We assume an audio file has been normalized to a fixed root mean square energy and segmented into a series of (possibly overlapping) time windows called *frames*. For each frame, a short time Fourier transform (STFT) is performed with a hamming window and 4 times zero-padding to get a power spectrum.

Spectral peaks are then detected by the peak detector described in [26]. Basically, there are two criteria that determine whether a power spectrum local maximum is labeled a peak. The first criterion is global: the local maximum should not be less than some threshold (e.g., 50 dB) lower than the global maximum of the spectrum. The second criterion is local: the local maximum should be locally higher than a smoothed version of the spectrum by at least some threshold (e.g., 4 dB). Finally, the peak amplitudes and frequencies are refined by quadratic interpolation [25].

Given this set of peaks, a set C of candidate F0s is generated. To facilitate computation, we do not consider the "missing fundamental" situation in this paper. Candidate F0 values are restricted to a range of $\pm 6\%$ in Hz (one semitone) of the frequency of an observed peak. We consider increments with a step size of 1% in Hz of the peak frequency. Thus, for each observed peak we have 13 candidate F0 values. In implementation, we can further reduce the search space by assuming F0s only occur around the five lowest frequency peaks, five highest amplitude peaks, and five locally highest amplitude peaks (peak amplitudes minus the smoothed spectral envelope). This gives at most $15 \cdot 13 = 195$ candidate F0s for each frame.

A naive approach to finding the best set of F0s would have to consider the power set of these candidates: 2^{195} sets. To deal with this issue, we use a greedy search strategy, which estimates F0s one by one. This greatly reduces the time complexity (for a complexity analysis see Section VI).

At each iteration, a newly estimated F0 is added to the existing F0 estimates until the maximum allowed polyphony is reached. Then, a postprocessor (Section IV) determines the best polyphony using a threshold base on the likelihood improvement as each F0 estimate is added. Finally, each frame's F0 estimates are refined using information from estimates in neighboring frames (see Section V).

III. Estimating F0s

This section describes how we approach steps 6 and 7 of the algorithm in Table I. Given a time frame presumed to contain N monophonic harmonic sound sources, we view the problem of estimating the fundamental frequency (F0) of each source as a Maximum Likelihood parameter estimation problem in the frequency domain

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\mathbf{O} \mid \boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{\theta} = \left\{F_0^1, \ldots, F_0^N\right\}$ is a set of N fundamental frequencies to be estimated, $\boldsymbol{\Theta}$ is the space of possible sets $\boldsymbol{\theta}$, and $\mathbf{O}$ represents our observation from the power spectrum.

We assume that the spectrum is analyzed by a peak detector, which returns a set of peaks. The observation to be explained is the set of peaks *and* the non-peak region of the spectrum.

We define the *peak region* as the set of all frequencies within d of an observed peak. The *non-peak region* is defined as the complement of the peak region. We currently define d as a musical quarter tone, which will be explained in Section III-B. Then, similar to [20] and [21], peaks are further categorized into normal peaks and spurious peaks. From the generative model point of view, a *normal* peak is defined as a peak that is generated by a harmonic of an F0. Other peaks are defined as *spurious* peaks, which may be generated by peak detection errors, noise, sidelobes, etc.

The peak region likelihood is defined as the probability of occurrence of the peaks, given an assumed set of F0s. The non-peak region likelihood is defined as the probability of *not* observing peaks in the non-peak region, given an assumed set of F0s. The peak region likelihood and the non-peak region likelihood act as a complementary pair. The former helps find F0s that have harmonics that explain peaks, while the latter helps avoid F0s that have harmonics in the non-peak region.

We wish to find the set $\boldsymbol{\theta}$ of F0s that maximizes the probability of having harmonics that could explain the observed peaks, and minimizes the probability of having harmonics where no peaks were observed. To simplify calculation, we assume independence between peaks and the non-peak region. Correspondingly, the likelihood is defined as the multiplication of two parts: the peak region likelihood and the non-peak region likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{peak region}}(\boldsymbol{\theta}) \cdot \mathcal{L}_{\text{non-peak region}}(\boldsymbol{\theta}). \tag{2}$$

The parameters of the models are learned from training data, which are summarized in Table II and will be described in detail in the following.

A. Peak Region Likelihood

Each detected peak k in the power spectrum is represented by its frequency f_k and amplitude a_k. Given K peaks in the spectrum, we define the peak region likelihood as

$$\begin{aligned} \mathcal{L}_{\text{peak region}}(\boldsymbol{\theta}) &= p(f_1, a_1, \ldots, f_K, a_K \mid \boldsymbol{\theta}) & (3)\\ &\approx \prod_{k=1}^{K} p(f_k, a_k \mid \boldsymbol{\theta}). & (4) \end{aligned}$$

Note that f_k, a_k and all other frequencies and amplitudes in this paper are measured on a logarithmic scale (musical semitones and dB, respectively).[1] This is done for ease of manipulation and accordance with human perception. Because frequency is calculated in the semitone scale, the distance between any two frequencies related by an octave is always 12 units. We adopt the general MIDI convention of assigning the value 60 to Middle $C(C4, 262$ Hz) and use a reference frequency of $A = 440$ Hz. The MIDI number for $A = 440$ Hz is 69, since it is 9 semitones above Middle C.

From (3) to (4), we assume[2] conditional independence between observed peaks, given a set of F0s. Given a harmonic sound, observed peaks ideally represent harmonics and are at integer multiples of F0s. In practice, some peaks are caused by inherent limitations of the peak detection method, non-harmonic resonances, interference between overlapping sound sources and noise. Following the practice of [20], we call peaks caused by harmonics *normal* peaks, and the others *spurious* peaks. We need different models for normal and spurious peaks.

For monophonic signal, there are several methods to discriminate normal and spurious peaks according to their shapes [27], [28]. For polyphonic signal, however, peaks from one source may overlap peaks from another. The resulting composite peaks cannot be reliably categorized using these methods. Therefore, we introduce a binary random variable s_k for each peak to represent that it is normal $(s_k = 0)$ or spurious $(s_k = 1)$, and consider both cases in a probabilistic way:

$$p\left(f_k, a_k|\boldsymbol{\theta}\right) = \sum_{s_k} p\left(f_k, a_k|s_k, \boldsymbol{\theta}\right) P\left(s_k|\boldsymbol{\theta}\right). \tag{5}$$

$P\left(s_k|\boldsymbol{\theta}\right)$ in (5) represents the prior probability of a detected peak being normal or spurious, given a set of F0s.[3] We would like to learn it from training data. However, the size of the space for $\boldsymbol{\theta}$ prohibits a creating data set with sufficient coverage. Instead, we neglect the effects of F0s on this probability and learn $P\left(s_k\right)$ to approximate $P\left(s_k|\boldsymbol{\theta}\right)$. This approximation is not only necessary, but also reasonable. Although $P\left(s_k|\boldsymbol{\theta}\right)$ is influenced by factors related to F0s, it is much more influenced by the limitations of the peak detector, nonharmonic resonances and noise, all of which are independent of F0s.

We estimate $P\left(s_k\right)$ from *randomly mixed chords*, which are created using recordings of individual notes performed by a variety of instruments (see Section VII-A for details). For each frame of a chord, spectral peaks are detected using the peak detector described in [26]. Ground-truth values for F0s are obtained by running YIN [29], a robust single F0 detection algorithm, on the recording of each individual note, prior to combining them to form the chord.

We need to classify normal and spurious peaks and collect their corresponding statistics in the training data. In the training data, we have the ground-truth F0s; hence, the classification becomes possible. We calculate the frequency deviation of each peak from the nearest harmonic position of the reported ground-truth F0s. If the deviation d is less than a musical quarter tone (half a semitone), the peak is labeled normal, otherwise spurious. The justification for this value is: YIN is a robust F0 estimator. Hence, its reported ground-truth F0 is within a quarter tone range of the unknown true F0, and its reported harmonic positions are within a quarter tone range of the true harmonic positions. As a normal peak appears at a harmonic position of the unknown true F0, the frequency deviation of the normal peak defined above will be smaller than a quarter tone. In our training data, the proportion of normal peaks is 99.3% and is used as $P\left(s_k = 0\right)$.

In (5), there are two probabilities to be modeled, i.e., the conditional probability of the normal peaks $p\left(f_k, a_k|s_k = 0, \boldsymbol{\theta}\right)$ and the spurious peaks $p\left(f_k, a_k|s_k = 1, \boldsymbol{\theta}\right)$. We now address them in turn.

1) Normal Peaks: A normal peak may be a harmonic of only one F0, or several F0s when they all have a harmonic at the peak position. In the former case, $p\left(f_k, a_k|s_k = 0, \boldsymbol{\theta}\right)$ needs only consider one F0. However, in the second case, this probability is conditioned on multiple F0s. This leads to a combinatorial problem we wish to avoid.

To do this, we adopt the assumption of binary masking [30], [31] used in some source separation methods. They assume the energy in each frequency bin of the mixture spectrum is caused by only one source signal. Here we use a similar assumption that each peak is generated by only one F0, the one having the largest likelihood to generate the peak:

$$p\left(f_k, a_k|s_k = 0, \boldsymbol{\theta}\right) \approx \max_{F_0 \in \boldsymbol{\theta}} p\left(f_k, a_k|F_0\right). \tag{6}$$

Now let us consider how to model $p\left(f_k, a_k|F_0\right)$. Since the kth peak is supposed to represent some harmonic of F_0, it is reasonable to calculate the harmonic number h_k as the nearest harmonic position of F_0 from f_k.

Given this, we find the harmonic number of the nearest harmonic of an F0 to an observed peak as follows:

$$h_k = \left[2^{f_k - F_0/12}\right] \tag{7}$$

where $[\cdot]$ denotes rounding to the nearest integer. Now the frequency deviation d_k of the kth peak from the nearest harmonic position of the given F0 can be calculated as

$$d_k = f_k - F_0 - 12\log_2 h_k. \tag{8}$$

To gain a feel for how reasonable various independence assumptions between our variables might be, we collected statistics on the randomly mixed chord data described in Section VII-A. Normal peaks and their corresponding F0s are detected as described before. Their harmonic numbers and frequency deviations from corresponding ideal harmonics are also calculated. Then the correlation coefficient is calculated for each pair of these variables. Table III illustrates the correlation coefficients between f_k, a_k, h_k, d_k, and F_0 on this data.

We can factorize $p\left(f_k, a_k|F_0\right)$ as

$$p\left(f_k, a_k|F_0\right) = p\left(f_k|F_0\right) p\left(a_k|f_k, F_0\right). \tag{9}$$

[1] FREQUENCY: MIDI number $= 69 + 12 \times \log_2$ (Hz/440); AMPLITUDE: dB $= 20 \times \log_{10}$ (Linear amplitude).

[2] In this paper, we use $\approx$ to denote "assumption."

[3] Here $P(\cdot)$ denotes probability mass function of discrete variables; $p(\cdot)$ denotes probability density of continuous variables.

TABLE III
CORRELATION COEFFICIENTS BETWEEN SEVERAL VARIABLES OF NORMAL PEAKS OF THE POLYPHONIC TRAINING DATA

	a	f	F_0	h	d
a	1.00	-0.78	**-0.11**	**-0.60**	-0.01
f	–	1.00	0.40	0.56	0.01
F_0	–	–	1.00	-0.41	-0.01
h	–	–	–	1.00	0.02
d	–	–	–	–	1.00

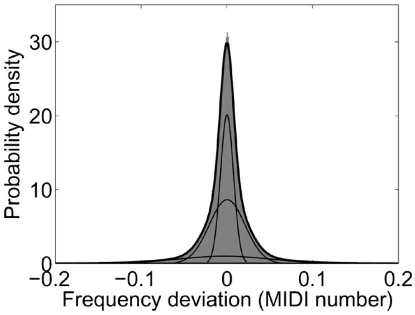


Fig. 1. Illustration of modeling the frequency deviation of normal peaks. The probability density (bold curve) is estimated using a Gaussian mixture model with four kernels (thin curves) on the histogram (gray area).

To model $p(f_k|F_0)$, we note from (8) that the relationship between the frequency of a peak f_k and its deviation from a harmonic d_k is linear, given a fixed harmonic number h_k. Therefore, in each segment of f_k where h_k remains constant, we have

$$p(f_k|F_0) = p(d_k|F_0) \tag{10}$$

$$\approx p(d_k) \tag{11}$$

where in (11), $p(d_k|F_0)$ is approximated by $p(d_k)$. This approximation is supported by the statistics in Table III, as the correlation coefficients between d and F_0 is very small, i.e., they are statistically independent.

Since we characterize $p(d_k)$ in relation to a harmonic, and we measure frequency in a log scale, we build a standard normalized histogram for d_k in relation to the nearest harmonic and use the same distribution, regardless of the harmonic number. In this paper, we estimate the distribution from the randomly mixed chords data set described in Section VII-A. The resulting distribution is plotted in Fig. 1.

It can be seen that this distribution is symmetric about zero, a little long tailed, but not very spiky. Previous methods [18], [20], [21] model this distribution with a single Gaussian. We found a Gaussian mixture model (GMM) with four kernels to be a better approximation. The probability density of the kernels and the mixture is also plotted in Fig. 1.

To model $p(a_k|f_k, F_0)$, we observe from Table III that a_k is much more correlated with h_k than F_0 on our data set. Also, knowing two of f_k, h_k and F_0 lets one derive the third value [as in (8)]. Therefore, we can replace F_0 with h_k in the condition

$$p(a_k|f_k, F_0) = p(a_k|f_k, h_k) = \frac{p(a_k, f_k, h_k)}{p(f_k, h_k)}. \tag{12}$$

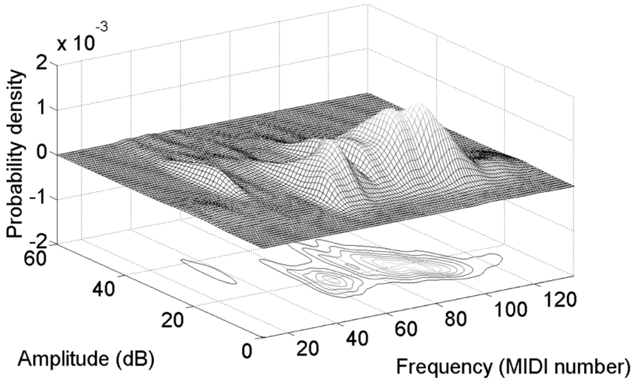


Fig. 2. Illustration of the probability density of $p(f_k, a_k|s_k = 1)$, which is calculated from the spurious peaks of the polyphonic training data. The contours of the density is plotted at the bottom of the figure.

We then estimate $p(a_k, f_k, h_k)$ using the Parzen window method [32], because it is hard to characterize this probability distribution with a parametric representation. An $11\ (\mathrm{dB}) \times 11\ (\mathrm{semitone}) \times 5$ Gaussian window with variance 4 in each dimension is used to smooth the estimate. The size of the window is not optimized but just chosen to make the probability density look smooth.

We now turn to modeling those peaks that were not associated with a harmonic of any F0.

2) Spurious Peaks: By definition, a spurious peak is detected by the peak detector, but is not a harmonic of any F0 in $\boldsymbol{\theta}$, the set of F0s. The likelihood of a spurious peak from (4) can be written as

$$p(f_k, a_k|s_k = 1, \boldsymbol{\theta}) = p(f_k, a_k|s_k = 1). \tag{13}$$

The statistics of spurious peaks in the training data are used to model (13). The shape of this probability density is plotted in Fig. 2, where a $11\ (\mathrm{semitone}) \times 9\ (\mathrm{dB})$ Gaussian window is used to smooth it. Again, the size of the window is not optimized but just chosen to make the probability density look smooth. It is a multi-modal distribution, however, since the prior probability of spurious peaks is rather small (0.007 for our training data), there is no need to model this density very precisely. Here a 2-D Gaussian distribution is used, whose means and covariance are calculated to be $(82.1, 23.0)$ and $\begin{pmatrix} 481.6 & -89.5 \\ -89.5 & 86.8 \end{pmatrix}$.

We have now shown how to estimate probability distributions for all the random variables used to calculate the likelihood of an observed peak region, given a set of F0s, using (3). We now turn to the non-peak region likelihood.

B. Non-Peak Region Likelihood

As stated in the start of Section III, the non-peak region also contains useful information for F0 estimation. However, how is it related to F0s or their predicted harmonics? Instead of telling us where F0s or their predicted harmonics should be, the non-peak region tells us where they should not be. A good set of F0s would predict as few harmonics as possible in the non-peak region. This is because if there is a predicted harmonic in the non-peak region, then clearly it was not detected. From the generative model point of view, there is a probability for each harmonic being or not being detected. Therefore, we define the non-peak region likelihood in terms of the probability of *not* detecting any harmonic in the non-peak region, given an assumed set of F0s.

We assume that the probability of detecting a harmonic in the non-peak region is independent of whether or not other harmonics are detected. Therefore, the probability can be written as the multiplication of the probability for each harmonic of each F0, as

$$\mathcal{L}_{\text{non-peak region}}(\boldsymbol{\theta}) \approx \prod_{F_0\in\boldsymbol{\theta}} \prod_{\substack{h\in\{1\cdots H\}\\ F_h\in\mathcal{F}_{\text{np}}}} 1-P\left(e_h=1|F_0\right) \quad (14)$$

where $F_h = F_0 + 12\log_2^h$ is the frequency (in semitones) of the predicted hth harmonic of F_0; e_h is the binary variable that indicates whether this harmonic is detected, $\mathcal{F}_{\text{np}}$ is the set of frequencies in the non-peak region, and H is the largest harmonic number we consider.

In the definition of the non-peak region in Section I-B, there is a parameter d controlling the size of the peak region and the non-peak region. It is noted that this parameter does not affect the peak-region likelihood, but only affects the non-peak region likelihood. This is because the smaller d is, the larger the non-peak region is and the higher the probability that the set of F0 estimates predicts harmonics in the non-peak region.

Although the power spectrum is calculated with an STFT and the peak widths (main lobe width) are the same in terms of Hz for peaks with different frequencies, d should not be defined as constant in Hz. Instead, d should vary linearly with the frequency (in Hz) of a peak. This is because d does not represent the width of each peak, but rather the possible range of frequencies in which a harmonic of a hypothesized F0 may appear. This possible range increases as frequency increases. In this paper, d is set to a musical quarter tone, which is 3% of the peak frequency in Hz. This is also in accordance with the standard tolerance of measuring correctness of F0 estimation.

Now, to model $P\left(e_h = 1|F_0\right)$. There are two reasons that a harmonic may not be detected in the non-peak region: First, the corresponding peak in the source signal was too weak to be detected (e.g., high frequency harmonics of many instruments). In this case, the probability that it is not detected can be learned from monophonic training samples.

Second, there is a strong corresponding peak in the source signal, but an even stronger nearby peak of another source signal prevents its detection. We call this situation *masking*. As we are modeling the non-peak region likelihood, we only care about the masking that happens in the non-peak region.

To determine when masking may occur with our system, we generated 100 000 pairs of sinusoids with random amplitude differences from 0 to 50 dB, frequency differences from 0 to 100 Hz and initial phase difference from 0 to 2π. We found that as long as the amplitude difference between two peaks is less than 50 dB, neither peak is masked if their frequency difference is over a certain threshold; otherwise, the weaker one is always masked. The threshold is 30 Hz for a 46-ms frame with a 44.1-kHz sampling rate. These are the values for frame size and sample rate used in our experiments.

For frequencies higher than 1030 Hz, a musical quarter tone is larger than $1030 \times 2^{1/24} = 30.2$ Hz. The peak region contains frequencies within a quarter tone of a peak, Therefore, if masking takes place, it will be in the peak region. In order to account for the fact that the masking region due to the fast Fourier transform (FFT) bin size (30 Hz) is wider than a musical quarter tone for frequencies under 1030 Hz, we also tried a definition of d that chose the maximum of either a musical quarter tone or 30 Hz: $d = \max(0.5 \text{ semitone}, 30 \text{ Hz})$. We found the results were similar to those achieved using the simpler definition of $d = 0.5$ semitone.

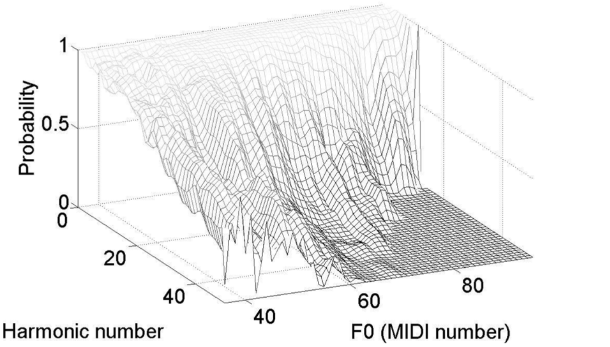


Fig. 3. Probability of detecting the hth harmonic, given the F0: $P\left(e_h = 1|F_0\right)$. This is calculated from monophonic training data.

Therefore, we disregard masking in the non-peak region. We estimate $P\left(e_h^n = 1|F_0\right)$, i.e., the probability of detecting the hth harmonic of F_0 in the source signal, by running our peak detector on the set of individual notes from a variety of instruments used to compose chords in Section VII-A. F0s of these notes are quantized into semitones. All examples with the same quantized F0 are placed into the same group. The probability of detecting each harmonic, given a quantized F0 is estimated by the proportion of times a corresponding peak is detected in the group of examples. The probability for an arbitrary F0 is then interpolated from these probabilities for quantized F0s.

Fig. 3 illustrates the conditional probability. It can be seen that the detection rates of lower harmonics are large, while those of the higher harmonics become smaller. This is reasonable since for many harmonic sources (e.g., most acoustic musical instruments) the energy of the higher frequency harmonics is usually lower. Hence, peaks corresponding to them are more difficult to detect. At the right corner of the figure, there is a triangular area where the detection rates are zero, because the harmonics of the F0s in that area are out of the frequency range of the spectrum.

IV. Estimating the Polyphony

Polyphony estimation is a difficult subproblem of multiple F0 estimation. Researchers have proposed several methods together with their F0 estimation methods [8], [17], [23].

In this paper, the polyphony estimation problem is closely related to the overfitting often seen with Maximum-Likelihood methods. Note that in (6), the F_0 is selected from the set of estimated F0s, $\boldsymbol{\theta}$, to maximize the likelihood of each normal peak. As new F0s are added to $\boldsymbol{\theta}$, the maximum likelihood will never decrease and may increase. Therefore, the larger the polyphony, the higher the peak likelihood is

$$\mathcal{L}_{\text{peak region}}\left(\hat{\boldsymbol{\theta}}^n\right) \leq \mathcal{L}_{\text{peak region}}\left(\hat{\boldsymbol{\theta}}^{n+1}\right) \quad (15)$$

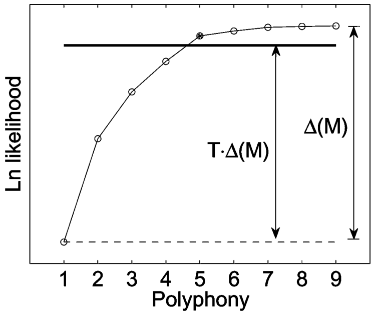


Fig. 4. Illustration of polyphony estimation. Log likelihoods given each polyphony are depicted by circles. The solid horizontal line is the adaptive threshold. For this sound example, the method correctly estimates the polyphony, which is 5, marked with an asterisk.

where $\hat{\boldsymbol{\theta}}^n$ is the set of F0s that maximize (2) when the polyphony is set to n. $\hat{\boldsymbol{\theta}}^{n+1}$ is defined similarly. If one lets the size of $\boldsymbol{\theta}$ range freely, the result is that the explanation returned would be the largest set of F0s allowed by the implementation.

This problem is alleviated by the non-peak region likelihood, since in (14), adding one more F0 to $\boldsymbol{\theta}$ should result in a smaller value $\mathcal{L}_{\text{non-peak region}}(\boldsymbol{\theta})$

$$\mathcal{L}_{\text{non-peak region}}\left(\hat{\boldsymbol{\theta}}^n\right) \geq \mathcal{L}_{\text{non-peak region}}\left(\hat{\boldsymbol{\theta}}^{n+1}\right). \qquad (16)$$

However, experimentally we find that the total likelihood $\mathcal{L}(\boldsymbol{\theta})$ still increases when expanding the list of estimated F0s

$$\mathcal{L}\left(\hat{\boldsymbol{\theta}}^n\right) \leq \mathcal{L}\left(\hat{\boldsymbol{\theta}}^{n+1}\right). \qquad (17)$$

Another method to control the overfitting is needed. We first tried using a Bayesian information criterion, as in [22], but found that it did not work very well. Instead, we developed a simple threshold-based method to estimate the polyphony N

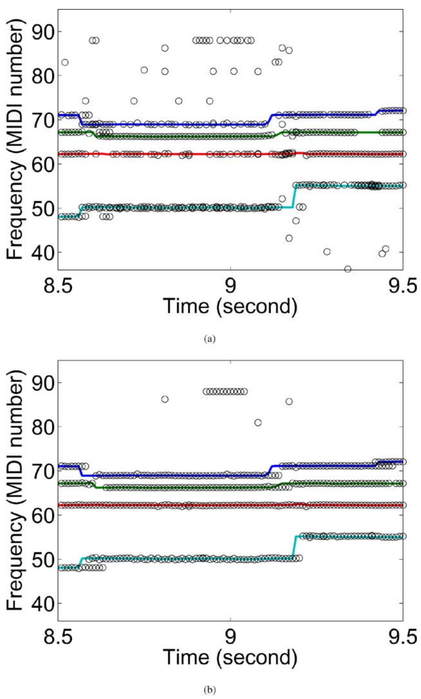
$$\begin{aligned} N = & \min_{1\leq n \leq M} n, \\ \text{s.t.} \quad & \Delta(n) \geq T \cdot \Delta(M) \end{aligned} \qquad (18)$$

, where $\Delta(n) = \ln \mathcal{L}(\hat{\boldsymbol{\theta}}^n) - \ln \mathcal{L}(\hat{\boldsymbol{\theta}}^1)$; M is the maximum allowed polyphony; T is a learned threshold. For all experiments in this paper, the maximum polyphony M is set to 9. T is empirically determined to be 0.88. The method returns the minimum polyphony n that has a value $\Delta(n)$ exceeding the threshold. Fig. 4 illustrates the method. Note that Klapuri [17] adopts a similar idea in polyphony estimation, although the thresholds are applied to different functions.

V. Postprocessing Using Neighboring Frames

F0 and polyphony estimation in a single frame is not robust. There are often insertion, deletion, and substitution errors [see Fig. 5(a)]. Since pitches of music signals are locally (on the order of 100 ms) stable, it is reasonable to use F0 estimates from neighboring frames to refine F0 estimates in the current frame. In this section, we propose a refinement method with two steps: remove likely errors and reconstruct estimates.

1) Step 1: Remove F0 estimates inconsistent with their neighbors.



(a)

(b)

Fig. 5. F0 estimation results before and after refinement. In both figures, lines illustrate the ground-truth F0s, circles are the F0 estimates. (a) Before refinement. (b) After refinement.

To do this, we build a weighted histogram W in the frequency domain for each time frame t. There are 60 bins in W, corresponding to the 60 semitones from C2 to B6. Then, a triangular weighting function in the time domain w centered at time t is imposed on a neighborhood of t, whose radius is R frames. Each element of W is calculated as the weighted frequency of occurrence of a quantized (rounded to the nearest semitone) F0 estimate. If the true polyphony N is known, the N bins of W with largest histogram values are selected to form a refined list. Otherwise, we use the weighted average of the polyphony estimates in this neighborhood as the refined polyphony estimate N, and then form the refined list.

2) Step 2: Reconstruct the non-quantized F0 values.

We update the F0 estimates for frame t as follows. Create one F0 value for each histogram bin in the refined list. For each bin, if an original F0 estimate (unquantized) for frame t falls in that bin, simply use that value, since it was probably estimated correctly. If no original F0 estimate for frame t falls in the bin, use the weighted average of the original F0 estimates in its neighborhood in this bin.

In this paper, R is set to nine frames (90 ms with 10-ms frame hop). This value is not optimized. Fig. 5 shows an example with the ground truth F0s and F0 estimates before and after this refinement. It can be seen that a number of insertion and deletion errors are removed, making the estimates more "continuous." However, consistent errors, such as the circles in the top middle part of Fig. 5(a), cannot be removed using this method.

It is noted that a side effect of the refinement is the removal of duplicate F0 estimates (multiple estimates within a histogram bin). This will improve precision if there are no unisons between sources in the data set, and will decrease recall if there are.

VI. Computational Complexity

We analyze the runtime complexity of the algorithm in Table I in terms of the number of observed peaks K and the maximum allowed polyphony M. We can ignore the harmonic number upper bound H and the number of neighboring frames R, because both these variables are bounded by fixed values.

The time of Steps 2 through 4 is bounded by a constant value. Step 5 is a loop over Steps 6 through 8 with M iterations. Steps 6 and 7 involves $|\mathcal{C}| = O(K)$ likelihood calculations of (2). Each one consists of the peak region and the non-peak region likelihood calculation. The former costs $O(K)$, since it is decomposed into K individual peak likelihoods in (4) and each involves constant-time operations. The latter costs $O(M)$, since we consider MH harmonics in (14). Step 9 involves $O(M)$ operations to decide the polyphony. Step 10 is a constant-time operation. Step 12 involves $O(M)$ operations. Thus, total runtime complexity in each single frame is $O(MK^2 + M^2K))$. If M is fixed to a small number, the runtime can be said to be $O(K^2)$.

If the greedy search strategy is replaced by the brute force search strategy, that is, to enumerate all the possible F0 candidate combinations, then Steps 5 through 8 would cost $O(2^K)$. Thus, the greedy approach saves considerable time.

Note that each likelihood calculation for (2) costs $O(K+M)$. This is a significant advantage compared with Thornburg and Leistikow's monophonic F0 estimation method [20]. In their method, to calculate the likelihood of a F0 hypothesis, they enumerate all associations between the observed peaks and the underlying true harmonics. The enumeration number is shown to be exponential in $K + H$. Although a MCMC approximation for the enumeration is used, the computational cost is still much heavier than ours.

VII. Experiments

A. Data Set

The monophonic training data are *monophonic note recordings*, selected from the University of Iowa website.[4] In total, 508 note samples from 16 instruments, including wind (flute), reed (clarinet, oboe, saxophone), brass (trumpet, horn, trombone, tuba), and arco string (violin, viola, bass) instruments were selected. They were all of dynamic "mf" and "ff" with pitches ranging from C2 (65 Hz, MIDI number 36) to B6 (1976 Hz, MIDI number 95). Some samples had vibrato. The polyphonic training data are *randomly mixed chords*, generated by combining these monophonic note recordings. In total 3000 chords, 500 of each polyphony from 1 to 6 were generated.

Chords were generated by first randomly allocating pitches without duplicates, then randomly assigning note samples of those pitches. Different pitches might come from the same instrument. These note samples were normalized to have the same root-mean-squared amplitude, and then mixed to generate chords. In training, each note/chord was broken into frames with length of 93 ms and overlap of 46 ms. A short time Fourier transform (STFT) with 4 times zero padding was employed on each frame. All the frames were used to learn model parameters.

The polyphony estimation algorithm was tested using 6000 *musical chords*, 1000 of each polyphony from 1 to 6. They were generated using another 1086 monophonic notes from the Iowa data set. These were of the same instruments, pitch ranges, etc., as the training notes, but were not used to generate the training chords. Musical chords of polyphony 2, 3, and 4 were generated from commonly used note intervals. Triads were major, minor, augmented, and diminished. Seventh chords were major, minor, dominant, diminished, and half-diminished. Musical chords of polyphony 5 and 6 were all seventh chords, so there were always octave relations in each chord.

The proposed multiple F0 estimation method was tested on ten real music performances, totalling 330 seconds of audio. Each performance was of a four-part Bach chorale, performed by a quartet of instruments: violin, clarinet, tenor saxophone, and bassoon. Each musician's part was recorded in isolation while the musician listened to the others through headphones. In testing, each piece was broken into frames with length of 46 ms and a 10-ms hop between frame centers. All the frames were processed by the algorithm. We used a shorter frame duration on this data to adapt to fast notes in the Bach chorales. The sampling rate of all the data was 44.1 kHz. A sample piece can be accessed through " http://music.cs.northwestern.edu/lab/research.php " under Section "Multi-pitch Estimation."

B. Ground-Truth and Error Measures

The ground-truth F0s of the testing pieces were estimated using YIN [29] on the single-instrument recordings prior to mixing recordings into four-part monaural recordings. The results of YIN were manually corrected where necessary.

The performance of our algorithm was evaluated using several error measures. In the *Predominant-F0 estimation* (Pre-F0) situation, only the first estimated F0 was evaluated [7]. It was defined to be correct if it deviated less than a quarter tone (3% in Hz) from any ground-truth F0. The estimation accuracy was calculated as the amount of correct predominant F0 estimates divided by the number of testing frames.

In the *Multi-F0 estimation* (Mul-F0) situation, all F0 estimates were evaluated. For each frame, the set of F0 estimates and the set of ground-truth F0s were each sorted in ascending order of frequency. For each F0 estimate starting from the lowest, the lowest-frequency ground-truth F0 from which it deviated less than a quarter tone was matched to the F0 estimate. If a match was found, the F0 estimate was defined to be correctly estimated, and the matched ground-truth F0 was removed from its set. This was repeated for every F0 estimate. After this process terminated, unassigned elements in either the estimate set or the ground-truth set were called errors. Given this, *Precision*, *Recall*, and *Accuracy* were calculated as

$$\text{Precision} = \frac{\#\text{cor}}{\#\text{est}} \quad \text{Recall} = \frac{\#\text{cor}}{\#\text{ref}} \tag{19}$$

$$\text{Accuracy} = \frac{\#\text{cor}}{\#\text{est} + \#\text{ref} - \#\text{cor}} \tag{20}$$

where #ref is the total number of ground truth F0s in testing frames, #est is the total number of estimated F0s, and #cor is the total number of correctly estimated F0s.

[4] http://theremin.music.uiowa.edu/

Octave errors are the most common errors in multiple F0 estimation. Here we calculate octave error rates as follows: After the matching process in Mul-F0, for each unmatched ground-truth F0, we try to match it with an unmatched F0 estimate after transcribing the estimate to higher or lower octave(s). *Lower-octave error rate* is calculated as the number of these newly matched F0 estimates after a higher octave(s) transcription, divided by the number of ground-truth F0s. *Higher-octave error rate* is calculated similarly.

For polyphony estimation, a mean square error (MSE) measure is defined as

$$\text{Polyphony-MSE} = \text{Mean}\left\{(P_{\text{est}} - P_{\text{ref}})^2\right\} \tag{21}$$

where P_{est} and P_{ref} are the estimated and the true polyphony in each frame, respectively.

C. Reference Methods

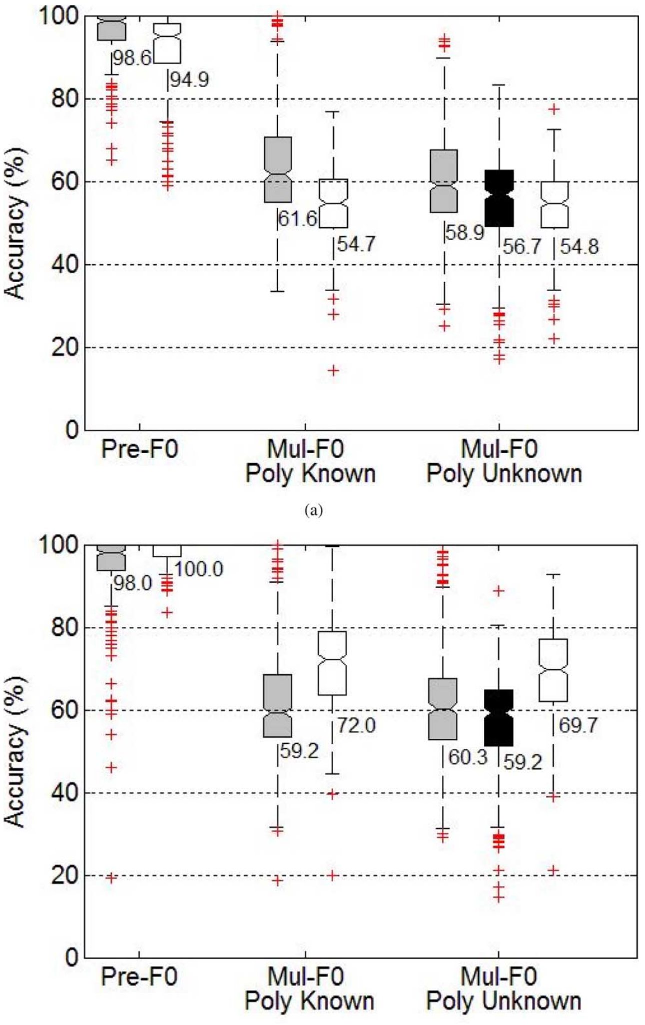
Since our method is related to previous methods based on modeling spectral peaks, it would be reasonable to compare our performance to the performance of these systems. However, [18]–[20] are all single F0 estimation methods. Although [21] is a multiple F0 estimation method, the computational complexity of the approach makes it prohibitively time-consuming, as shown in Section VI. Instead, our reference methods are the one proposed by Klapuri in [17] (denoted as “Klapuri06”) and the one proposed by Pertusa and Iñesta in [23] (denoted as “Pertusa08”). These two methods both were in the top 3 in the “Multiple Fundamental Frequency Estimation & Tracking” task in the Music Information Retrieval Evaluation eXchange (MIREX) in 2007 and 2008.[5]

Klapuri06 works in an iterative fashion by estimating the most significant F0 from the spectrum of the current mixture and then removing its harmonics from the mixture spectrum. It also proposes a polyphony estimator to terminate the iteration. Pertusa08 selects a set of F0 candidates in each frame from spectral peaks and generates all their possible combinations. The best combination is chosen according to their harmonic amplitudes and a proposed spectral smoothness measure. The polyphony is estimated simultaneously with the F0s. For both reference methods, we use the authors’ original source code and suggested settings in our comparison.

D. Multiple F0 Estimation Results

Results reported here are for the 330 seconds of audio from ten four-part Bach chorales described in Section VII-A. Our method and the reference methods are all evaluated once per second, in which there are 100 frames. Statistics are then calculated from the per-second measurements.

We first compare the estimation results of the three methods in each single frame without refinement using context information. Then we compare their results with refinement. For Klapuri06, which does not have a refinement step, we apply our context-based refinement method (Section V) to it. We think this is reasonable because our refinement method is quite general and not coupled with our single frame F0 estimation method. Pertusa08, has its own refinement method using information across frames. Therefore, we use Pertusa08’s own method. Since Pertusa08 estimates all F0s in a frame simultaneously, Pre-F0 is not a meaningful measure on this system. Also, Pertusa08’s original program does not utilize the polyphony information if the true polyphony is provided, so Mul-F0 Poly Known is not evaluated for it.



(a)

(b)

Fig. 6. F0 estimation accuracy comparisons of Klapuri06 (gray), Pertusa08 (black), and our method (white). In (b), Klapuri06 is refined with our refinement method and Pertusa08 is refined with its own method.

Fig. 6 shows box plots of F0 estimation accuracy comparisons. Each box represents 330 data points. The lower and upper lines of each box show 25th and 75th percentiles of the sample. The line in the middle of each box is the sample median, which is also presented as the number below the box. The lines extending above and below each box show the extent of the rest of the samples, excluding outliers. Outliers are defined as points over 1.5 times the interquartile range from the sample median and are shown as crosses.

As expected, in both figures the Pre-F0 accuracies of both Klapuri06’s and ours are high, while the Mul-F0 accuracies are much lower. Before refinement, the results of our system

[5] http://www.music-ir.org/mirex/

TABLE IV
Mul-F0 Estimation Performance Comparison, When the Polyphony is not Provided to the Algorithm

	Accuracy	Precision	Recall
Klapuri06	59.7±11.6	**86.1±9.6**	66.0±11.5
Pertusa08	57.3±11.4	84.6±13.5	63.7±9.6
Our method	**68.9±10.8**	82.7±8.1	**80.2±10.3**

are worse than Klapuri06's and Pertusa08's. Take Mul-F0 Poly Unknown as an example, the median accuracy of our method is about 4% lower than Klapuri06's and 2% lower than Pertusa08's. This indicates that Klapuri06 and Pertusa08 both gets better single frame estimation results. A nonparametric sign test performed over all measured frames on the Mul-F0 Poly Unknown case shows that Klapuri06 and Pertusa08 obtains statistically superior results to our method with p-value $p < 10^{-9}$ and $p = 0.11$, respectively.

After the refinement, however, our results are improved significantly, while Klapuri06's results generally stay the same and Pertusa08's results are improved slightly. This makes our results better than Klapuri06's and Pertusa08's. Take the Mul-F0 Poly Unknown example again, the median accuracy of our system is about 9% higher than Klapuri06's and 10% higher than Pertusa08's. A nonparametric sign test shows that our results are superior to both reference methods with $p < 10^{-25}$.

Since we apply our refinement method on Klapuri06 and our refinement method removes inconsistent errors while strengthening consistent errors, we believe that the estimation errors in Klapuri06 are more consistent than ours.

Remember that removing duplicates is a side effect of our postprocessing method. Since our base method allows duplicate F0 estimates, but the data set rarely contains unisons between sources, removing duplicates accounts for about 5% of Mul-F0 accuracy improvement for our method in both Poly Known and Unknown cases. Since Klapuri06 removes duplicate estimates as part of the approach, this is another reason our refinement has less effect on Klapuri06.

Fig. 6 shows a comparison of our full system [white boxes in (b)] to the Kapuri06 as originally provided to us [gray boxes in (a)] and Pertusa08's system [black box in (b)]. A nonparametric sign test on the Mul-F0 Poly Unknown case shows our system's superior performance was statistically significant with $p < 10^{-28}$.

Table IV details the performance comparisons of Mul-F0 Poly Unknown in the format of "Mean ± Standard deviation" of all three systems. All systems had similar precision, however Klapuri06 and Pertusa08 showed much lower accuray and recall than our system. This indicates both methods underestimate the number of F0s. This analysis is supported in our analysis of polyphony estimation.

E. Polyphony Estimation Results

Since polyphony estimation is a difficult task itself, we evaluated all three methods on this task. Among the 33 000 frames in Bach chorale test data, 29 687 had instruments sounding. Since all the pieces are quartets, and every instrument is active all along, the ground-truth polyphony is set to four for every frame with instruments sounding (we ignore the few frames that some performer ended or started a touch early).

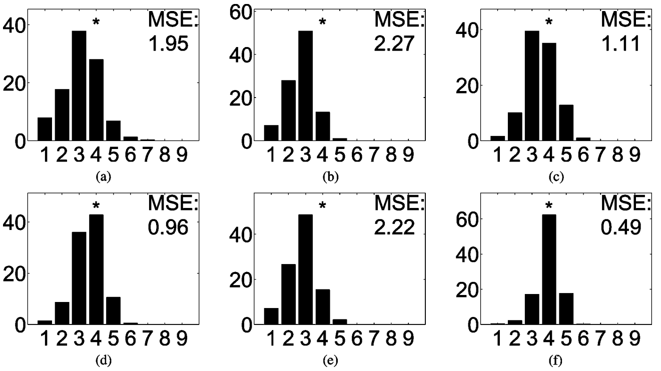


Fig. 7. Polyphony estimate histogram on the total 33 000 frames of the testing music pieces. X-axes represent polyphony. Y-axes represent the proportion of frames (%). The asterisk indicates the true polyphony. (a) Klapuri06. (b) Pertusa08. (c) Our method. (d) Klapuri06 with our refinement. (e) Pertusa08 with its own refinement. (f) Our method with our refinement.

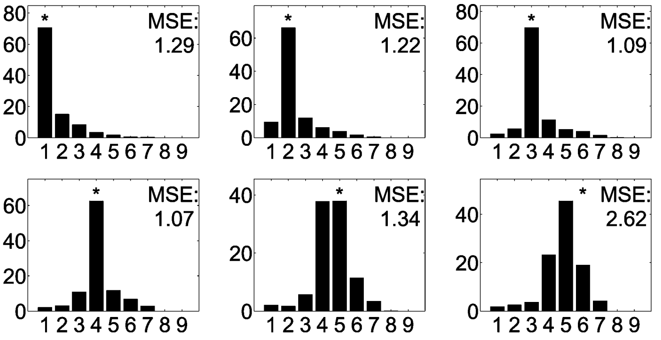


Fig. 8. Polyphony estimation histogram of musical chords with polyphony from 1 to 6. X-axes represent polyphony. Y-axes represent the proportion of frames (%). The asterisk indicates the true polyphony.

Fig. 7 shows the polyphony estimation histograms for all three methods without and with the refinement step. It can be seen that before refinement, all the methods tend to underestimate the polyphony. However, in both cases, our method obtains a better result with a lower MSE value than Klapuri06 and Pertusa08. Moreover, our refinement step improves the results for both Klapuri06 and our method, and finally our method obtains a symmetric histogram around the true polyphony as Fig. 7(f) shows.

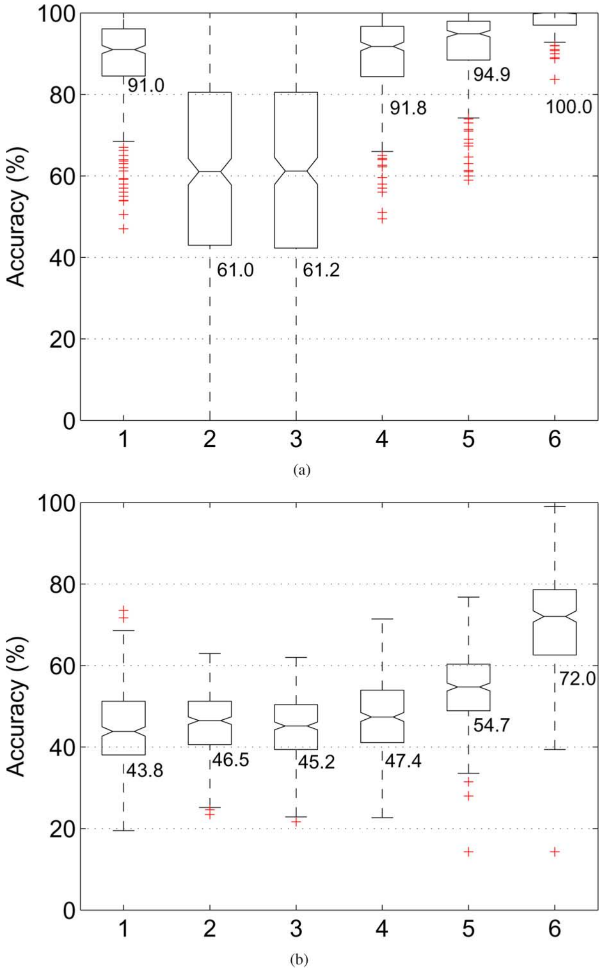
In order to evaluate our polyphony estimation method (Section IV without refinement) more comprehensively, we tested it on single frames of musical chords with different polyphony. Fig. 8 shows the results. It can be seen that in most examples of polyphony from 1 to 4, the system outputs the correct polyphony. However, for polyphony 5 and 6, the polyphony estimation results are not satisfying. One of the reason is that F0s with octave relations are difficult to estimate using our algorithm. In our data set, chords of polyphony 1, 2, 3, and 4 do not have octave relations. Each chord of polyphony 5 contains a pair of pitches related by an octave. This means 40% of the pitches are in an octave relation. Each chord of polyphony 6 contains two pairs, giving 66.7% of the pitches in an octave relation. Thus, the tendency to underestimate their polyphony is not surprising.

F. Individual Analysis of System Components

When a new approach is introduced, it may not always be clear which aspects of it contribute most strongly to its performance. We now investigate the effectiveness of different techniques that are used in our method: modeling peak frequencies and amplitudes, considering the possibility of spurious peaks, modeling the non-peak region, and refining F0 estimates using neighboring frames. In this experiment, we compare the F0 estimation accuracies with different system configurations:

- 1: Models peak frequency deviations with a single Gaussian, as in [18].
- 2: Models peak frequency deviations with a GMM model.
- 3: System 2 + models peak amplitudes with the non-parametric model in (12).
- 4: System 3 + considers the possibility of spurious peaks.
- 5: System 4 + models the non-peak region with (14).
- 6: System 5 + refines F0 estimates, as in Section V.

Box plots of F0 estimation accuracies of these systems when the true polyphony is provided are illustrated in Fig. 9. Again, each box represents 330 data points, corresponding to the 330 seconds of our testing pieces. For Pre-F0 results, systems except 2 and 3 are all higher than 90%. From System 2 to 3, the single Gaussian is replaced by a GMM to model the peak frequency deviation, which makes it possible to represent the tail of the distribution in Fig. 1. Therefore, the frequency restriction of each F0 estimate is loosened, and the accuracy of the predominant F0 estimate is lower. However, after adding the spurious peak model in System 4 and the non-peak region model in System 5, more restrictions are added to F0 estimates and the accuracy is improved. Finally, the F0 refinement technique improves the Pre-F0 median accuracy to 100.0%. We note that the predominant F0 estimate in a frame after the refinement may not be the same predominant F0 estimate as before, instead, it is the best predominant F0 estimate in the neighborhood, hence is more robust.



(a)

(b)

Fig. 9. F0 estimation accuracy of different system configurations of our method, when the true polyphony is provided. The x-axes are system configuration numbers. (a) Pre-F0. (b) Mul-F0.

For Mul-F0, the F0 estimation accuracy generally increases from System 1 to 6. There are three main improvements of the median accuracy: a 2.7% increase by replacing the single Gaussian with a GMM of modeling peak frequency deviations (System 1 to 2); a 7.3% increase by adding the non-peak region model (System 4 to 5); a 17.3% increase by F0 refinement (System 5 to 6). All of these improvements are statistically significant with $p < 10^{-8}$ in a nonparametric sign test. The only decrease occurs when adding the peak amplitude model (System 2 to 3). This indicates that the peak amplitude model parameters learned from randomly mixed chords are not suitable for the testing music pieces. In fact, when we train the peak likelihood parameters using five testing music pieces and test on all the ten pieces, System 2 achieves 46.0% (0.5% lower), while System 3 achieves Mul-F0 accuracy median of 49.5% (4.3% higher). This indicates two things: First, the peak frequency deviation model is well learned from randomly mixed chords; Second, the peak amplitude (timbre information) modeling is helpful only if the training data are similar to the testing data. However, due to the timbral variety of music, this situation can be rare. This is in accordance with Klapuri's observation in [16], [17], where he employs a spectral whitening technique to remove timbre information of both training and testing signals.

As octave errors are the most frequency errors in multiple F0 estimation, Fig. 10 shows the octave error rates of our systems. System 1 to 4 have much more lower-octave errors than higher-octave errors. This supports our claim that only modeling peaks will cause many lower octave errors. From System 4 to 5, lower-octave errors are significantly reduced because of the non-peak region model, as they have a small non-peak region likelihood. Lower-octave and higher-octave errors are then approximately balanced. It is noted that this balance is achieved automatically by our probabilistic model, while it is achieved by manual assignment of the balancing factor ρ in [19]. Finally, both octave errors are significantly reduced by the refinement.

We submitted our system to the "Multiple Fundamental Frequency Estimation & Tracking" task in MIREX 2009. "DHP2" is the system we described in this paper and "DHP1" is the multi-pitch tracking system built based on "DHP2". Both systems obtained good results. The results can be accessed at http://www.music-ir.org/mirex/2009/index.php/Multiple_Fundamental_Frequency_Estimation_&_Tracking_Results.

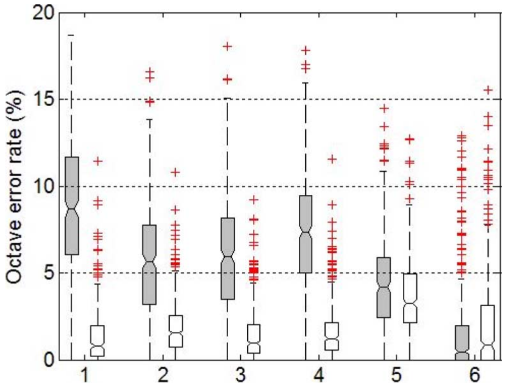


Fig. 10. Octave error (gray: lower-octave error, white: higher-octave error) rates of different system configurations of our method, when the true polyphony is provided. The x-axis is the system configuration number.

VIII. Conclusion

In this paper, we proposed a maximum-likelihood approach for multiple F0 estimation, where the power spectrum of a time frame is the observation and the F0s are the parameters to be estimated. The proposed method reduces the power spectrum into a peak region and a non-peak region, and the likelihood function is defined on both parts. The peak region likelihood is defined as the probability that a peak is detected in the spectrum given a set of F0s. The non-peak region likelihood is defined as the probability of not detecting any harmonics in the non-peak region. The two parts act as a complementary pair. To solve the combinatorial problem of simultaneously estimating F0s, we adopted an iterative estimation strategy to estimate F0s one by one. As expanding the number estimated F0s in each iteration, the total likelihood increases. We then proposed a polyphony estimation method by setting a threshold of the likelihood improvement. Finally, we proposed a refinement method to refine the F0 estimates using neighboring frames. This method removes a lot of inconsistent errors.

We tested the proposed approach on a corpus of ten instrumental recordings of J. S. Bach quartets. The results show the proposed approach outperforms two state-of-the-art algorithms on this data set on both F0 estimation and polyphony estimation. The polyphony estimation method is also tested on 6000 musical chords. Good results are obtained when there is no octave relation between pitches. It is noted that our system was trained using randomly mixed chords and monophonic notes, while tested on music pieces and musical chords. This indicates the generality of our system.

For sounds having octave-related pitches, the performance of our method will deteriorate, due to the binary masking assumption adopted in the peak region likelihood definition. Since octaves are the most common intervals encountered in music, this problem should be addressed in future work.

The current formulation limits the use of the method to harmonic sounds, but it should not be hard to extend it to quasi-harmonic sounds. The only change will occur in the calculation of the harmonic number of each peak.

Although using information from neighboring frames, the proposed method is still a F0 estimator rather than a F0 tracker. How to connect F0 estimates in adjacent frames and track them into F0 trajectories is a direction for future work.

Acknowledgment

The authors would like to thank A. Klapuri and A. Pertusa for generously providing source code for [17] and [23], respectively, for comparison. We also thank the reviewers, whose comments greatly improved this paper.

References

[1] G. E. Poliner and D. P. W. Ellis, “A discriminative model for polyphonic piano transcription,” *EURASIP J. Adv. Signal Process.*, vol. 2007, pp. 9–9.

[2] J. Woodruff, Y. Li, and D.-L. Wang, “Resolving overlapping harmonics for monaural musical sound separation using pitch and common amplitude modulation,” in *Proc. Int. Conf. Music Inf. Retrieval (ISMIR)*, 2008, pp. 538–543.

[3] N. Orio, S. Lemouton, and D. Schwarz, “Score following: State of the art and new developments,” in *Proc. Conf. New Interfaces for Musical Expression (NIME)*, 2003, pp. 36–41.

[4] M. Davy, S. J. Godsill, and J. Idier, “Bayesian analysis of western tonal music,” *J. Acoust. Soc. Amer.*, vol. 119, no. 4, pp. 2498–2517, 2006.

[5] E. Vincent and M. D. Plumbley, “Efficient Bayesian inference for harmonic models via adaptive posterior factorization,” *Neurocomputing*, vol. 72, no. 1-3, pp. 79–87, 2008.

[6] K. Kashino and H. Murase, “A sound source identification system for ensemble music based on template adaptation and music stream extraction,” *Speech Commun.*, vol. 27, pp. 337–349, 1999.

[7] M. Goto, “A real-time music scene description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals,” *Speech Commun.*, vol. 43, no. 4, pp. 311–329, 2004.

[8] H. Kameoka, T. Nishimoto, and S. Sagayama, “A multipitch analyzer based on harmonic temporal structured clustering,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 982–994, Mar. 2007.

[9] S. Saito, H. Kameoka, K. Takahashi, T. Nishimoto, and S. Sagayama, “Specmurt analysis of polyphonic music signals,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 3, pp. 639–650, Mar. 2008.

[10] J.-L. Durrieu, G. Richard, and B. David, “Singer melody extraction in polyphonic signals using source separation methods,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2008, pp. 169–172.

[11] V. Emiya, R. Badeau, and B. David, “Multipitch estimation of quasi-harmonic sounds in colored noise,” in *Proc. 10th Int. Conf. Digital Audio Effects (DAFx)*, 2007, pp. 93–98.

[12] G. Reis, N. Fonseca, and F. Ferndandez, “Genetic algorithm approach to polyphonic music transcription,” in *Proc. IEEE Int. Symp. Intell. Signal Process.*, 2007, pp. 321–326.

[13] R. Meddis and L. O’Mard, “A unitary model of pitch perception,” *J. Acoust. Soc. Amer.*, vol. 102, no. 3, pp. 1811–1820, 1997.

[14] T. Tolonen and M. Karjalainen, “A computationally efficient multipitch analysis model,” *IEEE Trans. Speech Audio Process.*, vol. 8, pp. 708–716, 2000.

[15] A. de Cheveigné and H. Kawahara, “Multiple period estimation and pitch perception model,” *Speech Commun.*, vol. 27, pp. 175–185, 1999.

[16] A. Klapuri, “Multiple fundamental frequency estimation based on harmonicity and spectral smoothness,” *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 804–815, Nov. 2003.

[17] A. Klapuri, “Multiple fundamental frequency estimation by summing harmonic amplitudes,” in *Proc. Int. Conf. Music Inf. Retrieval (ISMIR)*, 2006, pp. 216–221.

[18] J. Goldstein, “An optimum processor theory for the central formation of the pitch of complex tones,” *J. Acoust. Soc. Amer.*, vol. 54, pp. 1496–1516, 1973.

[19] R. C. Maher and J. W. Beauchamp, “Fundamental frequency estimation of musical signals using a two-way mismatch procedure,” *J. Acoust. Soc. Amer.*, vol. 95, pp. 2254–2263, 1993.

[20] H. D. Thornburg and R. J. Leistikow, “A new probabilistic spectral pitch estimatior: Extract and MCMC-approximate strategies,” *Lecture Notes in Computer Science*, vol. 3310/2005, pp. 41–60, 2005.

[21] R. J. Leistikow, H. Thornburg, J. O. Smith, and J. Berger, “Bayesian identification of closely-spaced chords from single-frame STFT peaks,” in *Proc. 7th Int. Conf. Digital Audio Effects (DAFx)*, 2004, pp. 228–233.

[22] Z. Duan and C. Zhang, “A maximum likelihood approach to multiple fundamental frequency estimation from the amplitude spectrum peaks,” in *Proc. Neural Inf. Process. Syst. (NIPS) Workshop Music, Brain, Cognition*, 2007.

[23] A. Pertusa and J. M. Iñesta, “Multiple fundamental frequency estimation using Gaussian smoothness,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2008, pp. 105–108.

[24] C. Yeh, A. Roebel, and X. Rodet, “Multiple fundamental frequency estimation of polyphonic music signals,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing(ICASSP)*, 2005, pp. 225–228.

[25] J. O. Smith and X. Serra, “PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation,” in *Proc. Int. Comput. Music Conf. (ICMC)*, 1987.

[26] Z. Duan, Y. Zhang, C. Zhang, and Z. Shi, “Unsupervised single-channel music source separation by average harmonic structure modeling,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 4, pp. 766–778, May 2008.

[27] X. Rodet, “Musical sound signal analysis/synthesis: Sinusoidal + residual and elementary waveform models,” in *Proc. IEEE Time-Frequency and Time-Scale Workshop (TFTS’97)*, 1997.

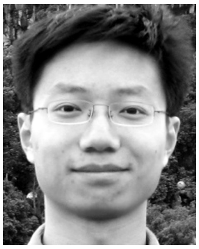
[28] A. Röbel and M. Zivanovic, “Signal decomposition by means of classification of spectral peaks,” in *Proc. Int. Comput. Music Conf. (ICMC)*, 2004.

[29] A. de Cheveigné and H. Kawahara, “Yin, a fundamental frequency estimator for speech and music,” *J. Acoust. Soc. Amer.*, vol. 111, pp. 1917–1930, 2002.

[30] Ö Yılmaz and S. Rickard, “Blind separation of speech mixtures via time-frequency masking,” *IEEE Trans. Signal Process.*, vol. 52, no. 7, pp. 1830–1847, Jul. 2004.

[31] M. H. Radfar and R. M. Dansereau, “Single-channel speech separation using soft mask filtering,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2299–2310, Nov. 2007.

[32] E. Parzen, “On the estimation of a probability density function and the mode,” *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.



Zhiyao Duan (S’09) was born in Henan, China, in 1983. He received the B.E. and M.S. degrees from Tsinghua University, Beijing, China, in 2004 and 2008, respectively. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL.

His research interests lie primarily in the interdisciplinary area of signal processing and machine learning toward audio information retrieval applications, including source separation, multiple pitch estimation, score alignment, etc.

Bryan Pardo (M’07) received the M.Mus. degree in jazz studies in and the Ph.D. degree in computer science in from the University of Michigan, Ann Arbor, in 2001 and 2005, respectively.

He is an Assistant Professor in the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, with appointments in the Music Cognition Program and the Center for Technology and Social Behavior. He has developed speech software for the Speech and Hearing, The Ohio State University, statistical software for SPSS and worked as a machine learning Researcher for General Dynamics. While finishing his Ph.D. degree, he taught in the Music Department of Madonna University. When he’s not programming, writing, or teaching, he performs throughout the United States on saxophone and clarinet at venues such as Albion College, the Chicago Cultural Center, the Detroit Concert of Colors, Bloomington Indiana’s Lotus Festival, and Tucson’s Rialto Theatre.

Changshui Zhang (M’02) received the B.S. degree in mathematics from Peking University, Beijing, China, in 1986 and the M.S. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, in 1989 and 1992, respectively.

In 1992, he joined the Department of Automation, Tsinghua University, and is currently a Professor. His interests include pattern recognition, machine learning, etc. He has authored more than 200 papers. He is currently an associate editor of the *Pattern Recognition* journal. He is also a member of the Standing Council of the Chinese Association of Artificial Intelligence.