IMPROVING POLYPHONIC MELODY EXTRACTION BY DYNAMIC PROGRAMMING BASED DUAL F0 TRACKING

Vishweshwara Rao and Preeti Rao

Department of Electrical Engineering,
Indian Institute of Technology Bombay
Mumbai, India
{vishu, prao}@ee.iitb.ac.in

ABSTRACT

The suitability of optimal path finding methods for vocal melody extraction in polyphonic music is well recognized since they combine local pitch strength and temporal smoothness considerations in a global sense. However, when such single-F0 tracking systems are applied to sound mixtures in which pitched accompaniment is of comparable strength to the singing voice, they suffer from irrecoverable degradations. In this study we investigate the use of an optimal path finding method that is allowed to dynamically track multiple F0 paths, specifically two, through the F0 candidate space. It is shown that when such a system is applied to typical polyphonic mixtures with vocal solo the melodic information is indeed retrieved. Audio examples are available at http://www.ee.iitb.ac.in/daplab/DualF0TrackingResults_DAFx

1. INTRODUCTION

Automatic melody extraction is an area of research that has received increased attention over the past decade. The majority of melody extraction algorithms in recent literature by-and-large adhere to the framework depicted in Figure 1. Here, initially a short-time, usually spectral, feature set is extracted from the input polyphonic audio signal. This is then input to a multi-F0 estimation block whose goal is to estimate candidate F0 and associated pitch salience/strength values. The melody identification stage attempts to identify a trajectory through the F0 candidate-time space that most likely represents the melody of the song, manifested as the pitch-curve of the lead melodic instrument, usually the singing voice for popular music. The voicing detection block identifies segments of audio in which the lead melodic instrument is present.

This study is mainly focused on improvements to the melody identification stage. In this stage most algorithms exploit F0 salience values, from the multi-F0 estimation stage, and impose smoothness constraints to identify the melodic trajectory. Two approaches have been widely used. The first involves finding an optimal path through the F0 space over time by dynamically combining F0 salience values and smoothness constraints [1]-[4]. All systems use optimal path finding methods either based on the Viterbi algorithm [5] or dynamic programming (DP) [6] to generate the final melodic trajectory. The second approach applies variants of the partial tracking (PT) algorithm, used in classical sinusoidal modeling [7], to forming multiple F0 trajectories/contours through the F0 candidate space over time [8]-[11]. The only criterion usually used in linking an F0 candidate to a 'track' is the frequency proximity of the candidate to the last tracked F0. The salience of the F0 is used to evaluate track strength and is not used as such in the linking process except sometimes in the case of when multiple tracks are competing for a single F0 candidate [9]. The final melodic contour is usually chosen as that track with the greatest salience /energy. Hereafter methods that use the first approach are referred to as DP and methods that use the second are referred to as PT.

Apart from finding the optimal path in terms of the defined cost functions, a further advantage of DP over PT is that it combines the trajectory forming and melodic contour identification in one, computationally-efficient, *global* framework i.e. a black box that outputs a single F0 contour given suitably defined local and smoothness costs. PT, on the other hand, first forms trajectories using *local* frequency proximity and subsequently identifies melodic 'tracks'. This last is not a trivial task [8] as the track-space can be densely populated especially in polyphonic music audio. Here multiple trajectories may be formed through the instrument F0 contours and their (sub) multiples since typically there will be F0 candidates at these values output by the multi-F0 module.

One condition under which both DP and PT may output an (partially) incorrect melody is when a strong, pitched accompanying instrument is also present i.e. low signal-to-accompaniment ratio (SAR). For such signals most multi-F0 modules would output an accompaniment F0 candidate with a salience comparable to the melodic F0 candidate, especially if the accompaniment has several, strong partials in the same frequency band as the melodic instrument. The melody identification stage then may incorrectly identify segments of the accompaniment pitch contour as the melody [8]. For PT based approaches the recovery of the actual melodic tracks may still be possible by the extraction of timbrally motivated features; based on the assumption that the melodic track is formed but not identified. DP, on the other hand, is forced to output only a single, possibly 'confused', contour.

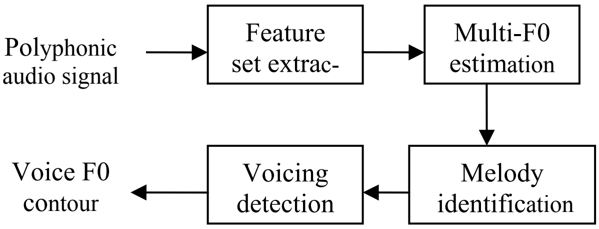

Figure 1: *Block diagram of a typical melody extraction system*

The reduction in pitch accuracy of a DP-based melody extractor, due to a strong, pitched accompanying instrument, may be alleviated if DP is extended to tracking multiple F0 contours simultaneously. This can be achieved by performing DP over a node-space as opposed to an F0 candidate space, where a node represents an ordered group of F0 candidates. The additional F0 members of the node help to better deal with the accompanying pitched instrument(s). The output would thus contain multiple F0 contours, which would then have to be examined for melodic contour identification. Although such a system is similar to the PT approach, in that the trajectory forming and melody identification are separate steps, the resulting trajectory-space formed as a result of node-based DP is significantly sparser than when using PT apart from being optimal cost.

In this study we propose such a system that tracks two simultaneous F0 contours (hereafter referred to as the dual-F0 system). We restrict ourselves to tracking only two pitches simultaneously on the realistic assumption that there is only one relatively dominant melodic or pitched instrument present at a given time besides the voice. Section 2 describes the implementation of the complete system. In Section 3 the single and dual-F0 systems are comparatively evaluated on a test data set of two-sound mixtures that exemplify the signal degradations that result in reduced pitch accuracy of the single-F0 tracking system. A simple post-processing method to correct errors peculiar to the dual-F0 system is also described. The last section presents the conclusions and directions for future study.

2. SYSTEM DESCRIPTION

The proposed system block diagram is shown in Figure 2. The system comprises of four modules. The first two modules (Sinusoid ID and F0 candidate selection) constitute the multi-pitch analysis that determines multiple F0s per frame. The next two modules (node forming and node tracking) describe the tracking stage in which F0 pairs are formed and are jointly tracked through the F0 candidate v/s time space. Parameter values for the algorithm are available in Table 1.

2.1. Multi-pitch analysis

The proposed system uses a method of extracting multiple F0 candidates, very similar to our previous melody extraction algorithm [12], which demonstrated high pitch tracking accuracies in the audio melody extraction task at the recent Music Information Retrieval Evaluation eXchange (MIREX 2008)[1]. The inputs to this system are the frequency locations and amplitudes of the local sinusoids/partials extracted from the feature extraction module, which detects sinusoids in the magnitude spectrum, obtained from a fixed-frame signal analysis and a high-resolution STFT, using a main-lobe matching criterion and refines the sinusoid frequency and amplitude estimates using parabolic interpolation.

For the dual-F0 case it is required to reliably detect both source F0s for each analysis frame in order to ensure that the melodic F0 is not lost. Different methods of estimating two distinct F0s simultaneously from a monophonic pitch detection algorithm (PDA) were studied by de Cheveigné [13], [14]. The first of these is to extend a single-F0 algorithm to estimate two F0 e.g. identifying the largest and second largest peak in the autocorrelation function. The next method is to use an iterative "estimate-cancel-estimate" method that first estimates a dominant F0 and then removes its effect from the signal and re-estimates the dominant F0. The third method is to jointly estimate 2 F0s from the signal. This usually involves searching a two-dimensional space of F0 pairs for that pair that minimizes/maximizes some function of 2 F0s. It was found [14] that the third method, while computationally expensive, had a lower error rate than the second method, which in turn performed better than the first method.

2.1.1. Implementation

Our system uses a computationally-efficient method of reliably extracting multiple F0 candidates, without having to resort to the iterative or joint estimation approach, by separating the F0 candidate detection from the salience computation along the lines of [15]. Probable candidate locations are first identified as sub-multiples of the frequencies of well-formed sinusoids i.e. those having a sinusoidality (*S*) greater than 0.8. Candidates that do not lie within the F0 search range (from 70 to 1120 Hz, spanning four octaves) are erased.

For each of the above detected candidates their corresponding salience is computed as the normalized Two-Way Mismatch (TWM) error [16]. In a previous study we had found that the melodic (voice) F0 candidate was detected with significantly higher salience, in the presence of strong, spectrally sparse, tonal interferences, by searching for local minima from the TWM error curve for different trial F0 [1]. The TWM PDA falls under the category of harmonic matching (monophonic) PDAs that are based on the frequency domain matching of a measured spectrum with an ideal harmonic spectrum. However, unlike typical harmonic matching algorithms that maximize the energy at the expected ideal harmonic locations, the TWM PDA minimizes a spectral mismatch error that is a particular combination of an individual partial's frequency deviation from the ideal harmonic location and its relative strength. Rather than dependence on individual harmonic strength, which is the case with most 'harmonic-sieve' based methods [17], we found the TWM error values to be more dependent on the harmonic spectral spread. This has two advantages. F0s belonging to the singing voice, which is known to have a large harmonic spread, are expected to have lower TWM errors i.e. better salience. Additionally even strong (loud) pitched accompaniment which is spectrally sparse, such as guitar, piano, pitched percussion, will have lower salience.

The overall TWM error (Err_{TWM}), for a given trial F0 (f), is a weighted sum of two errors, the predicted-to-measured error ($Err_{p \to m}$) and the measured-to-predicted error ($Err_{m \to p}$), as shown in Equation 1.

Table 1: *Multi-pitch analysis parameters.*

Parameter	Value
Frame length	40 ms
Hop	10 ms
Lower limit on F0	70 Hz
Upper limit on F0	1120 Hz
Upper limit on spectral content	5000 Hz
NFFT	8192
Single-F0 TWM param. (p, q, r & ρ)	0.5, 1.4, 0.5 & 0.1
Dual-F0 TWM param. (p, q, r & ρ)	0.5, 1.4, 0.5 & 0.33

[1] Detailed results are available at http://www.music-ir.org/mirex/2008/index.php/Audio_Melody_Extraction_Results

$$Err_{TWM}(f) = \frac{Err_{p\to m}(f)}{N} + \rho\frac{Err_{m\to p}(f)}{M} \tag{1}$$

where N and M are the number of predicted and measured harmonics respectively and ρ is the weighting factor. $Err_{p\to m}$ is based on the mismatch between each harmonic in the predicted sequence and its nearest neighbor in the measured partials while $Err_{m\to p}$ is based on the frequency difference between each partial in the measured sequence and its nearest neighbor in the predicted sequence. Both of these share the same form. $Err_{p\to m}$ is defined below.

$$Err_{p\to m} = \sum_{n=1}^{N}\left[\frac{\Delta f_n}{(f_n)^p} + \left(\frac{a_n}{A_{\max}}\right)\left(q\frac{\Delta f_n}{(f_n)^p} - r\right)\right] \tag{2}$$

where f_n and a_n are the frequency and magnitude of a single predicted harmonic. Δf_n is the difference, in Hz, between this harmonic and its nearest neighbor in the list of measured partials. A_{max} is the magnitude of the strongest measured partial. Thus an amplitude weighted penalty is applied to a normalized frequency error between measured and predicted partials for the given trial F0. p, q, and r are independent parameters. Note that here we use a value $\rho = 0.1$. This gives lesser weight to $Err_{m\to p}$ and leads to Err_{TWM} being almost the same as $Err_{p\to m}$ since $Err_{m\to p}$ for single-F0 values will be unreliable in the presence of harmonics of another pitched source.

Next, the F0 candidates are sorted in ascending order of their individual TWM errors (Err_{TWM}). Weaker candidates (having higher TWM error) that lie in the close vicinity (25 cents) of a stronger candidate are erased from the list of possible F0 candidates. Only the top 10 candidates and their corresponding normalized TWM error values from the final list are chosen for further processing.

2.1.2. Evaluation

In a preliminary evaluation of our multi-pitch analysis system we used complex tone mixtures made available by Tolonen [18] in which two harmonic complexes, added at different amplitude ratios of 0, 3, 6 and 10 dB, whose F0s are spaced a semitone apart (140 and 148.3 Hz) are considered. It was shown that the discernibility of a peak at the weaker F0 candidate in an enhanced summary autocorrelation function (ESACF) progressively gets worse; while at 6 dB it is visible as a shoulder peak, at 10 dB it cannot be detected [18]. For our study an evaluation metric of 'percentage presence' was defined as the percentage of frames that an F0 candidate is found within 15 cents of the ground truth F0. We found that for all mixtures (0, 3, 6 and 10 dB) both F0s (140 and 148.3 Hz) were always detected by our system i.e. percentage presence = 100%. This indicates that the F0 presence of the relatively weak source is clearly signaled in the TWM error curve as obtained by us.

2.2. Multi-pitch tracking

2.2.1. Related literature

There is relatively sparse literature on joint tracking of F0 combinations. The system proposed by Li and Wang [4] was specifically designed to track the F0 of the singing voice in polyphonic audio. This system used HMMs to track pitch states. Each pitch state could be represented by a 0, 1 or 2-pitch hypothesis. The 2-pitch hypothesis introduced to deal with the interference from concurrent pitched sounds. Here, however all possible pairs of locally salient F0 candidates are considered. This may lead to the irrelevant and unnecessary tracking of an F0 and its (sub)-multiple, which often tend to have similar local salience as the true pitch. Also when two pitches are tracked the first pitch is always considered to be the voice pitch since it is considered to be the dominant pitch whenever present.

2.2.2. Implementation

Our system extends the single-F0 tracking DP algorithm to track ordered F0 pairs called nodes. If we consider all possible pairs of F0 candidates the combinatory space will become very large (Number of permutations of F0 pairs formed from 10 F0 candidates is ${}^{10}P_2 = 90$) and tracking will be computationally intensive. More importantly, we may end up tracking an F0 and its (sub)-multiples, as mentioned before. Our method to overcome this is to explicitly prohibit the pairing of harmonically related F0s during node generation. Specifically, two local F0 candidates (f_1 and f_2) will be paired only if

$$\min_k\left(\left|f_1 - k.f_2\right|\right) > T; \quad k.f_2 \in \left[F_{low}, F_{high}\right] \tag{3}$$

where $k.f_2$ represents all possible multiples and sub-multiples of f_2, T is the harmonic relationship threshold and F_{low} and F_{high} are the lower and upper limit on the F0 search range (see Table 1). Using a low threshold (T) of 5 cents does not allow F0s to be paired with their multiples but allows pairing of two source F0s that are playing an octave apart, which typically suffer from slight detuning especially if one of the F0 sources is the singing voice.

The measurement cost of a node is defined as the jointly estimated TWM error of its constituent F0 candidates [19]. In the interest of computational efficiency the joint TWM error is computed as shown below

$$Err_{TWM}(f_1,f_2) = \frac{Err_{p\to m}(f_1)}{N_1} + \frac{Err_{p\to m}(f_2)}{N_2} + \rho\frac{Err_{m\to p}(f_1,f_2)}{M} \tag{4}$$

where N_1 and N_2 are the number of predicted partials for f_1 and f_2 resp. and M is the number of measured partials. The first two terms in Equation 4 will have the same values as computed during the single F0 TWM error computation (Equation 1). Only the last term i.e. the mismatch between all measured partials and the predicted partials of both F0s (f_1 and f_2), has to be computed. Note that here we use a larger value of ρ (0.33) than before. This is done so as to reduce octave errors by increasing the weight of $Err_{m\to p}$ thereby ensuring that Err_{TWM} for the true F0 pair is lower than that of the pair that contains either of their respective (sub)-multiples.

The smoothness costs between nodes are the sum of smoothness costs between the constituent F0 candidates given by

$$W(p,p') = 1 - e^{\frac{-(\log_2(p') - \log_2(p))^2}{2\sigma}} \tag{5}$$

where p and p' are the ordered F0 candidates of nodes in the previous and current frames. A value of $\sigma = 0.1$ results in a function that assigns very low penalties to pitch transitions below 2 semitones [20]. Larger rates of pitch transition, in the 10 ms frame interval chosen in this work, are improbable, even during rapid singing pitch modulations, and are penalized accordingly.

A globally optimum path is then computed through the node-time space using the DP algorithm (hereafter referred to as the dual-F0 tracking system). Two pitch contours are then output.

Table 2: *Statistics of testing dataset.*

Category	Description	Vocal (sec)	Total (sec)
1	One pitched sound always present	24.1	26.1
2	0 pitched sounds may be present	25.2	30.4
3	F0 Collisions may occur	20.4	21.6
TOTAL		**70.1**	**78.1**

3. MELODY EXTRACTION EXPERIMENT

In this section the performance improvement of the proposed system (dual-F0) over the previous melody extraction algorithm (single-F0) is demonstrated on test data that comprises of cases for which the SARs between the singing voice and a pitched, accompanying instrument/another singing voice are very low.

3.1. Data Description

The data has been divided into three categories, as shown in Table 2. The first category contains mixes of real singing voice signals and real *harmonium* signals at 0 dB SAR. In these mixes, at any given time instant, there is at least one pitched sound present. Category 2 consists of a set of mixes of real singing voice signals and synthetic organ, and a set of mixes of two real singing voice signals. Here a complication is introduced that the instrumental note boundaries very often occur simultaneously with voice note boundaries, which are marked by unvoiced utterances. This leads to a case of zero pitched sounds being present at certain instants. The former set was specifically created for the source separation experiment in [21] but is also well suited for our evaluation. In this set the instrumental accompaniment is always playing the melody but at an octave higher than the voice. In the latter set, obtained from multi-track, studio-recorded, Indian film music, one voice is singing the melody while the other is ‘harmonizing’ with the first voice. All the mixes in this category are again at 0 dB SAR. Category 3 introduces a further complication of F0 collisions between the singing voice and instrument signals. This category contains 0 dB SAR mixtures of excerpts of real singing voice and real *harmonium* signals from multi-track recordings of actual North Indian classical vocal performances. The individual monophonic tracks were obtained by ensuring acoustic isolation between the instrument-performing artists by spreading them out on the same stage with considerable distance between them.

The statistics of each of the categories is shown in Table 2. Here total duration refers to the length of the entire audio and vocal duration refers to the duration for which sung voiced utterances are present.

3.2. Experiment and Results

A fixed set of PDA analysis parameters was used across the experiments (see Table 1). Note that our system does not place any restrictions on the relative pitch ranges of the two melodic sources nor does it impose specific rules on the kind of pitch transitions allowed. Additionally, no discretization of the melody in terms of note event [2], [8] is used since the melody in Indian-classical music is a continuously varying curve rather than a sequence of note events [20].

Table 3: *Percentage presence of melodic and accompanying ground truth F0 in candidate list output by multi-F0 module.*

Category	Percentage presence (%)	
	Voice F0	Instrument F0
1	99.7	98.5
2	98.4	97.5
3	96.6	99.2

Table 4: *Pitch accuracies (PA & CA) for test dataset using single and dual-F0 tracking.*

Category		Single-F0	Dual-F0	
			Best contour	Overall
1	PA (%)	57.0	97.4	97.4
	CA (%)	58.5	98.4	98.4
2	PA (%)	48.0	80.4	91.7
	CA (%)	55.1	83.2	94.0
3	PA (%)	52.5	66.3	85.0
	CA (%)	53.0	67.3	90.0

In all cases the ground truth voice pitch is computed from the clean voice tracks then hand corrected for voicing errors. Only valid ground-truth values i.e. for frames in which a pitched utterance is present, are used for evaluation.

The multi-pitch extraction part of our system is separately evaluated in terms of percentage presence, in the F0 candidate list, of the instrument and voice ground truth pitches. Results for this experiment are given in Table 3. Clearly the true F0 candidates for both sound sources are being detected with a high degree of accuracy.

The evaluation metrics used for melody extraction were pitch accuracy (PA) and chroma accuracy (CA) [17]. PA is defined as the percentage of voiced frames for which the pitch has been correctly detected i.e. within 50 cents of a ground-truth pitch. CA is the same as PA except that octave errors are forgiven. Results of the experiment are given in Table 4. Here the above metrics have been computed for the single-F0 tracking and the dual-F0 tracking system. Note that since no decision is currently being made about which of the 2 contours (in the form of one ordered pair of F0s per frame) output by the dual-F0 tracking system is the vocal contour, the results report that contour with the higher accuracy with respect to vocal pitch ground truth, and also present the overall accuracy as a measure of the correct pitch at a given instant being tracked by *at least* one of the two contours. In all cases the overall accuracy of the dual-F0 tracking system is seen to be higher than that of the single-F0 tracking system.

3.3. Discussion

3.3.1. Category 1

We observe, from the results presented in Table 4, that there is no difference between the accuracies of the best contour and the overall accuracy for this category of signals. This indicates that the best contour has faithfully tracked the voice F0 through the entire signal. Also the improvement over the single-F0 tracker is significant. To illustrate this point consider the contours output by the single and dual-F0 tracking systems for a single file in this category (See Figure 3). This file contains a mix of a female voice singing lyrics (i.e. both voiced and unvoiced sounds) with a *harmonium* signal that is playing notes with successively increasing pitch. In each plot, the ground truth of the singing voice (thin line) and the *harmonium* (dashed line) are also plotted. The gaps in the thin curve indicate unvoiced utterances. The thick curve represents the tracked contour. In Figure 3(a), we can see that the single-F0 tracker misses large parts of the voice pitch contour, during which it incorrectly tracks the *harmonium* contour. Figure 3(b) shows that contour 1 of the dual-F0 tracking system faithfully follows the voice pitch while Figure 3(c) shows that contour 2 faithfully tracks the *harmonium* pitch.

We observe from Figure 3(a) and (b) that during unvoiced utterances, an F0 pair (node) that consists of the *harmonium* F0 with a spurious F0 candidate is tracked. However, when the next voiced utterance occurs the tracked node pair consists of the *harmonium* and voice F0 again. The spurious candidate may be related to the *harmonium* F0 by a ratio of small integer numbers; as such candidates are usually available for pairing. The smoothness cost constraints will bias the system towards tracking such candidates that are in the neighborhood of the F0s of adjacent voiced utterances. The tracking of the spurious candidate does not degrade the pitch accuracy of the system since only the ground truth pitch of known voiced utterances are used in the computation of the evaluation metrics.

3.3.2. Category 2

From rows 3 and 4 of Table 4 we observe that the pitch tracking accuracies of the best tracked contour as well as the overall accuracies for the dual-F0 tracking system are again significantly higher than the single-F0 system output contour. However the overall accuracy values of the dual-F0 tracking system are higher than the best contour accuracies. This is due to the occurrence of 'switching' between the actual voice and instrument pitches across the two contours in the dual-F0 tracking system. This 'switching' is found to occur when the instrument note change occurs simultaneously with an unvoiced (un-pitched) sung utterance. All files in this category have frequent occurrences of the above situation as illustrated by the example of Figure 4. This example is a mix of a male voice singing lyrics and a synthetic organ. The convention for the different contours is the same as for Figure 3. Figure 4(a), (b) and (c) indicate the contour output by the single-F0 tracking system, contour 1 and contour 2 of the dual-F0 tracking system respectively.

The co-incident gaps in the thin and dashed contours indicate segments when no pitched sound is present. Figure 4(a) indicates that the output of the single-F0 tracking system is again 'confused' between the F0s of the two sources. However, even the dual-F0 output contours (Figure 4(b) and 4(c)) show similar degradation. It can be seen that contour 1 of the dual-F0 tracking system tracks the first note of the voice but then 'switches' to the organ F0 while the reverse happens for contour 2.

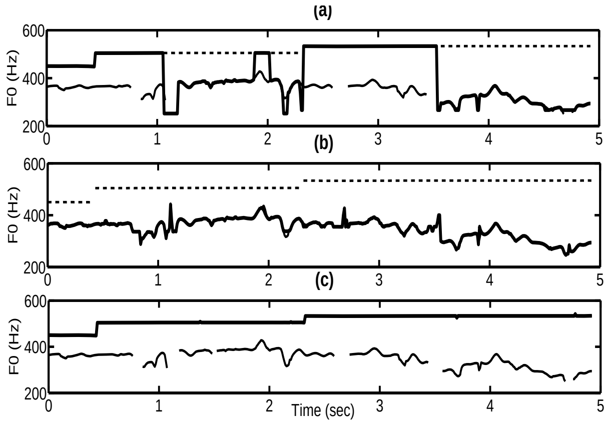

Figure 3: *Extracted F0 contours (thick) v/s ground truth F0s voice (thin) and harmonium (dashed) for (a) single-F0 tracking, (b) dual-F0 tracking: contour 1 and (c) contour 2 for an example from Category 1.*

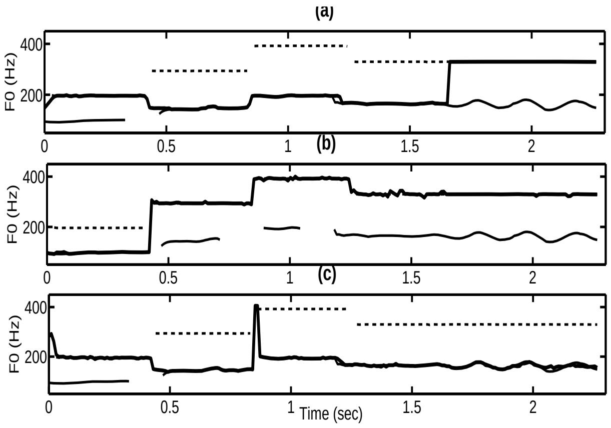

Figure 4: *Extracted F0 contours (thick) v/s ground truth F0s voice (thin) and organ (dashed) for (a) single-F0 tracking, (b) dual-F0 tracking: contour 1 and (c) contour 2 for an example from Category 2.*

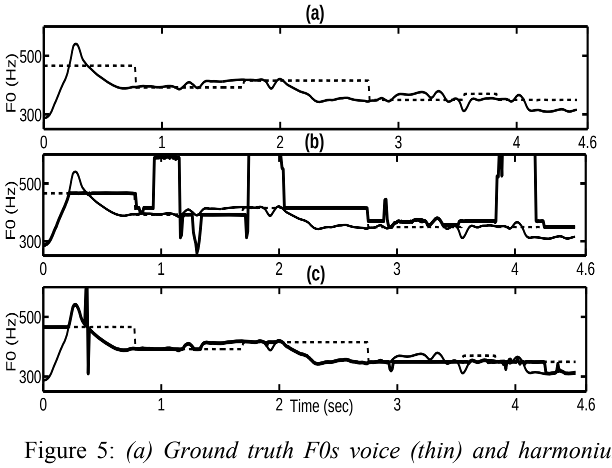

Figure 5: *(a) Ground truth F0s voice (thin) and harmonium (dashed) v/s (b) extracted F0 contours (thick) dual-F0 tracking: contour 1 and (c) contour 2 for an example from Category 3*

The current system cannot ensure that the contours will remain faithful to their respective sound sources across regions in which no clear pitched sound exists. Even if a zero-pitch hypothesis was made during these regions it would be difficult to ensure faithfulness, especially if the next note of the different source rather than the same source is closer to the previous note of a sound source. Further, it is seen occasionally that the slight detuning required for the correct clustering of pitches for the DP node formation does not always hold in the octave separated mixture. In such cases, spurious candidates are tracked instead as can be seen by the small fluctuations in the output contours of the dual-F0 tracking system (Figure 4(b) and (c)). Such fine errors do not occur in the cases of vocal harmony tracking.

3.3.3. Category 3

From rows 5 and 6 of Table 4 we can see that although the best contour accuracy of the dual-F0 tracking system is significantly better than the accuracy for the single-F0 tracking system, the overall accuracy of the dual-F0 system is still significantly higher than the former. This indicates that some ‘switching’ has taken place. Here F0 collisions are an additional complication. To illustrate this problem consider Figure 5 (b) and (c), which show the ground truth voice and instrument F0s along with the dual-F0 system output for a voice and *harmonium* mix from this category. For clarity, we have avoided plotting the single-F0 system output pitch contour in Figure 5(a), which now only shows the voice and *harmonium* ground truth values.

Figure 5(a) brings out a peculiarity of Indian classical music that causes F0 collisions to be a frequent rather than a rare occurrence. In this genre of music the *harmonium* accompaniment is meant to reinforce the melody sung by the singer. There is no score present as each vocal performance is a complete improvisation. So the instrumentalist attempts to follow the singer’s pitch contour as best he/she can. Since the *harmonium* is a keyed instrument, it cannot mimic the finer graces and ornamentation that characterize Indian classical singing but attempts to follow the steady held voice notes. This pitch following nature of the *harmonium* pitch is visible as the dashed contour following the thin contour in Figure 5(a).

At the locations of *harmonium* note change, the *harmonium* F0 intersecting with the voice F0 is similar to the previous case during unvoiced utterances when instead of two F0s only one true F0 is present. Here the contour tracking the *harmonium* will in all probability start tracking some spurious F0 candidates. During these instances the chances of switching are high since when the voice moves away from the *harmonium* after such a collision, the pitch-proximity based smoothness cost may cause the present contour to continue tracking *harmonium* while the contour tracking the spurious candidate may start tracking the voice F0.

Cases of the voice crossing a steady *harmonium* note should not usually result in a switch for the same reason that switching occurred in the previous case. The smoothness cost should allow the contour tracking *harmonium* to continue tracking *harmonium*. However the first collision, which is an example of voice F0 cross steady *harmonium* F0, causes a switch. This happened because of multiple conditions being simultaneously satisfied. The crossing is rapid and takes place exactly between the analysis time instants, the *harmonium* and voice F0 candidates are present on either side of the crossing but slightly deviated from their correct values due to the rapid pitch modulation. As Indian classical singing is replete with such rapid, large pitch modulations such a situation may not be a rare occurrence.

Table 5: *Pitch accuracies (PA & CA) of best contour before and after switching correction and overall accuracy for Category 2*

Category 2	PA (%)	CA (%)
Before post processing	80.4	83.2
After switching correction	89.4	90.7
Overall accuracy	91.7	94.0

3.4. Switching correction

From the previous results it has been shown that when short silences are present simultaneously for both sound sources (category 2) or when the F0 of the two sources ‘collide’ (category 3) individual contours tracking the F0s of either source may ‘switch’ over to tracking the F0s of the other source. This leads to lesser values of best contour accuracies when compared to the overall accuracies though they are still significantly higher than the single-F0 tracking accuracies. One simple solution to this problem proposed here is applicable when the F0 contours of the melodic and accompanying instruments do not collide (category 2). Often in western music, for the mixture of the tonal accompaniment and the melody to sound pleasing, their respective pitches must be musically related. Further, as opposed to Indian classical music, western (especially pop) music does not display such rapid pitch modulations. As a result, F0 collisions most often do not occur. This is also the case with musical harmony and duet songs.

With the above knowledge we implement switching correction by forcing one of the two F0 contours to always be higher or lower, in pitch, than the other F0 contour. To make the initial decision about which contour is lower/higher than the other we use a majority voting rule across the entire contour. From Table 5 we find that by applying the above correction the best contour PA and CA values of the dual-F0 tracking output for category 2 (row 2) come closer to the overall accuracy values (row 3).

4. CONCLUSION

In the context of melody extraction for vocal performances, it was found that a system that uses a dynamic programming framework for melody identification results in a single, degraded melodic contour when a strong, pitched accompanying instrument is present. This degradation is caused by the incorrect identification of the instrument pitch as the melody. In order to enable the recovery of the actual melodic contour it is proposed to extend the use of DP to tracking multiple pitch contours simultaneously. Specifically, a system that dynamically tracks F0-candidate pairs, generated by imposing specific harmonic relation-related constraints, is proposed to alleviate the above degradations. It is also possible to increase the number of F0s dynamically tracked further if there is more than one loud, pitched accompanying instrument. However, we have found in several cases of singing voice in polyphony that tracking just the one extra pitch is sufficient to retrieve the vocal pitch information. This can be explained by the fact that at any given time, there is usually not more than one accompanying instrument of comparable strength co-occurring with the singing voice.

It is found that when the proposed system is evaluated on mixtures of melodic singing voice and one loud pitched instrument and also cases of vocal harmony i.e. mixtures of two singing voices, the melodic voice pitch is tracked with increased accuracy by at least one of the contours at any given instant. This is an improvement over the previous single-F0 tracking system where the voice pitch was unrecoverable during pitch errors.

The proposed system does not make a decision on which of the 2 output F0 contours (or their sub-segments) belong to the singing voice. However, preliminary experiments have validated the performance of a vocal segment detection system that detects the relative instability of the voice pitch contours (via the frequency fluctuations of the harmonics) as compared to keyed instrument notes and uses it to label F0 contour segments as vocal or instrumental [22]. A problem pending investigation is that of F0 collisions. Such collisions, found to occur frequently in Indian classical music, induce contour switching and also the same pitch values have to be assigned to both contours during extended collisions. The latter condition can be achieved by pairing F0 candidates with themselves. But an indication of when such an exception should be made is required. We propose to investigate the use of predictive models of F0 contours, similar to those used for sinusoidal modeling in polyphony [23], and also possibly musicological rules to detect F0 collisions.

5. ACKNOWLEDGMENTS

The authors thank Mr. D. B. Biswas and Dr. S. Rao of the National Centre for the Performing Arts (NCPA), Mumbai, India for the multi-track Indian classical music data and Mr. Kailash Kher for the multi-track Indian film music data.

6. REFERENCES

[1] V. Rao and P. Rao, “Vocal melody detection in the presence of pitched accompaniment using harmonic matching methods,” in *Proc. of the 11th Intl. Conf. Digital Audio Effects (DAFx-08)*, Espoo, Finland, 2008.

[2] M. Rynänen and A. Klapuri, “Automatic transcription of melody, bass line and chords in polyphonic music,” *Comput. Music J.*, vol. 32, no. 3, pp. 72–86, Fall 2008.

[3] H. Fujihara et. al. “F0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and Viterbi search,” in *Proc. IEEE Intl. Conf. Audio, Speech Sig. Process.*, Toulouse, France, 2006.

[4] Y. Li and D. Wang “Separation of Singing Voice From Music Accompaniment for Monoaural Recordings,” *IEEE Trans. Audio, Speech Lang. Process.*, vol. 15, no. 4, pp. 1475–1487, May 2007.

[5] G. Forney, “The viterbi algorithm,” *Proc. of the IEEE*, vol. 61, no. 3, pp. 268 – 278, 1973.

[6] H. Ney, “Dynamic Programming Algorithm for Optimal Estimation of Speech Parameter Contours,” *IEEE Trans. Systems, Man and Cybernetics,* vol. SMC-13, no. 3, pp. 208–214, April 1983.

[7] J. McAulay and T. Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *IEEE Trans. Acoust., Speech, Sig. Process.*, vol. ASSP-34, no. 4, pp. 744–754, 1986.

[8] R. Paiva, T. Mendes and A. Cardoso, “Melody detection in polyphonic music signals,” *Comput. Music J.*, vol. 30, no. 4, pp. 80–98, Winter 2006.

[9] M. Goto, “A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real world audio signals,” *Speech Comm.*, vol. 43, pp. 311–329, 2004.

[10] K. Dressler, “An Auditory streaming approach to melody extraction,” in *MIREX Audio Melody Extraction Contest Abstracts*, Victoria, Canada, 2006.

[11] P. Cancela, “Tracking melody in polyphonic audio,” in *MIREX Audio Melody Extraction Contest Abstracts*, Philadelphia, 2008.

[12] V. Rao and P. Rao “Melody extraction using harmonic matching,” in *MIREX Audio Melody Extraction Contest Abstracts*, Philadelphia, 2008.

[13] A. de Cheveigné, “Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain model cancellation model of auditory processing,” *J. Acoust. Soc. Amer.*, vol. 93, no. 6, pp. 3271–3290, June 1993.

[14] A. de Cheveigné and H. Kawahara, “Multiple period estimation and pitch perception model,” *Speech Comm.*, vol. 27, no. 3, pp. 175-185, 1999.

[15] P. Cano, “Fundamental frequency estimation in the SMS analysis,” in *Proc. of COST G6 Conf. on Digital Audio Effects 1998*, Barcelona, Spain, 1998.

[16] R. Maher and J. Beauchamp, “Fundamental Frequency Estimation of Musical Signals using a Two-Way Mismatch Procedure,” *J. Acoust. Soc. Amer.*, vol. 95, no. 4, pp. 2254–2263, Apr. 1994.

[17] G. Poliner, et. al., “Melody transcription from music audio: Approaches and evaluation,” *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 15, no. 4, pp. 1247–1256, May 2007.

[18] T. Tolonen and M. Karjalainen, “A computationally efficient multipitch analysis model,” *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 708–716, Nov. 2000.

[19] R. Maher, “Evaluation of a method for separating digitized duet signals,” *J. Audio Eng. Soc.*, vol. 38, no. 12, pp. 956–979, 1990.

[20] A. Bapat, V. Rao and P. Rao, “Melodic contour extraction of Indian classical vocal music,” in *Proc. Intl. Workshop on Artificial Intelligence and Music (Music-AI ‘07)*, Hyderabad, India, Jan. 2007.

[21] 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 2004.

[22] V. Rao, S. Ramakrishnan and P. Rao, “Singing voice detection in polyphonic music using predominant pitch,” in *Proc. of Intl. Conf. on Spoken Lang. Process.*, Brighton, UK, Sept. 2009. (Accepted for publication).

[23] M. Lagrange, S. Marchand and J-B Rault, “Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds,” *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 15, no. 5, pp. 1625–1634, July 2007.