AUTUMN: A General Pitch-Extraction Wave-to-MIDI Transcription System

Kevin J. Di Filippo, Andrew Horner, Eric Fung, Jenny Lim, and Lydia Ayers

Department of Computer Science, Hong Kong University of Science and Technology
[kevind/horner/efung/jennylin/ayers]@cs.ust.hk

Abstract

Automatic Music Transcription is an area of active research at the intersection of Computer Science and Music, of interest for both theoretical and practical applications. We propose two Wave-to-MIDI transcription systems: Basic AUTUMN and V-AUTUMN. Both systems achieve their goal by performing window-by-window frequency-domain pitch extraction. Minimal assumptions are made on the content of the input signal. Listening tests show that AUTUMN performs acceptably well, with test subjects giving the system a rating of roughly 5 on a 9-point scale across a comprehensive range of musical examples. Quantitative analysis shows that AUTUMN can achieve success rates as high as 95%, with under 5% false-hit rates. Stability is the most important remaining problem to address.

1 Introduction

Automatic music transcription (AMT) is currently an area of active investigation in the fields of computer science and musical studies. The ultimate goal of AMT is to produce a complete symbolic representation of a musical performance based exclusively on the acoustic signal of that performance, without the need for human intervention. In particular, such a system would determine pitch values, note durations, instrumentation, lyrics, dynamics, and tempo markings; and would infer key signatures, time signatures, and phrasing. Such a system is considered to be currently unattainable and remains an open problem.

Research in AMT is instead currently focussed on exploring targeted subsets of full AMT. There is a wide range of applications that require only a partial solution to the AMT problem: music information retrieval querying and indexing, cellular phone ringtone transcription, and karaoke scoring systems, to list only a few. Therefore, although these applications can by definition be served by complete AMT, suitably focussed partial AMT systems offer widely-applicable solutions.

This paper contributes to one such area of partial AMT: that of acoustic to symbolic transcription. More precisely, this paper proposes a method of automatically performing pitch extraction from an acoustic signal of arbitrary musical content, while preserving the temporal (rhythmic)

characteristics of that signal. The only assumption on the content of the input acoustic signal is that it consists primarily of musical information, where we define “musical information” as audio information that can be represented as a series of zero or more simultaneous pitches. Some format limitations have been imposed purely for practical convenience: the algorithm only recognises PCM (Wave) audio format, and the input must be single-channel. However, these restrictions may be lifted without loss of generality. The system has been named *AUTUMN*, a phonetisation of “Automatic Wave TO MIDI.”

Groundwork for pitch extraction was laid by Klapuri (1999), who explored the determination of pitch in an acoustic signal containing a single note. As in many pitch extraction systems, spectral analysis lies at the core of the extraction method (Klapuri 2003). Specifically, in Klapuri’s approach, the acoustic signal was divided into multiple time windows; the pitch was independently estimated in each of these windows by performing harmonic grouping; and the final pitch was determined by combining these estimates. Later time-windowing methods explicitly relaxed the monophonic constraint and boosted performance using statistical methods (e.g., Goto 2001). These early time-windowing approaches were necessarily limited to single-event signals: the known time-invariance encourages accurateness in such methods and facilitates robustness mechanisms in the face of noise.

A second paradigm for specialised pitch determination relies on knowledge of the source instrument that produced the signal. For instance, Hainsworth and Macleod (2001) attempted to transcribe only bass-line notes, explicitly noting and making use of several characteristics typical to the bass lines of compositions, thus improving accuracy by effectively limiting the search and output spaces. Alternatively, instrument information can be used in a more direct capacity. For example, Sieger and Twefik (1997) introduced the concept of an “instrument dictionary,” a database containing spectral “snapshots” of a variety of instruments, and determined pitch by performing a window-by-window comparison between the acoustic signal of the input music file and the snapshots of the database. Such knowledge-focused approaches typically perform well in their domain, but suffer from a lack of generality.

In contrast to methods that gain stability at the cost of generality, other window-by-window paradigms exist that do *not* output a time-averaged pitch (Tolonen 2000). In such algorithms, pitches are reported on a window-by-window basis, with no attempt at smoothing by performing time-averaging. These “true” window-by-window algorithms gain the ability to distinguish note-events with a temporal resolution of the window-size, but at a potential sacrifice in stability. It is unclear in the literature exactly how severe this loss of stability is. For example, in Tolonen’s work, meaningful quantitative results are only available for time-invariant (single-event) signals, although qualitative results (e.g., rising-pitch vowel sounds) suggest that the stability may be acceptable in some situations. More recent methods regain stability by performing variants of time-averaging, but in a less naïve, more comprehensive way than the single-event methods (Martins and Ferreira 2002; Monti and Sandler 2002). Indeed, the methods presented in this paper are similar to these two methods (and closer specifically to Martins’), although details vary significantly.

Lastly, Klapuri *et al.* (2001) proposed a related variant on the traditional spectral-analysis pitch determination paradigm. Rather than directly attempt to determine the pitches from the spectrum, the authors attempted to first determine the number of voices present in the window. Pitch determination was subsequently based in part on the estimated number of present voices. Voice estimation was formulated as a noise-reduction problem. It is not clear from results whether this approach represents an improvement over the more traditional, heuristic methods presented above.

2 Basic AUTUMN

We now turn our attention to the methodologies proposed by this paper.

2.1 Overview

Figure 1 contains a block diagram of the basic Automatic Music Transcription system proposed by this paper. In short, it is a window-by-window spectral analyser that determines notes by harmonic grouping. This section provides a high-level overview of the system. Sections 2.2 to 2.4 describe details of each subsystem.

Basic AUTUMN works as follows. The input to the system is a raw acoustic signal. The signal is first converted from the time domain to the frequency domain by applying a Discrete Fourier Transform (DFT) on a series of overlapping temporal windows that spans the duration of the entire signal, which results in a set of “spectral snapshots,” each snapshot representing the frequency spectrum of the signal over a specific period of time. The results of the DFT are then processed to make them suitable for pitch extraction. Two processes occur at this stage. Firstly, the spectra of each window are independently normalised according to the total energy present in the window. This facilitates thresholding in the note-determination stage.

Secondly, short-time averaging over several windows is performed to improve detection stability. This averaging is performed in a sliding-window fashion to avoid loss of temporal resolution.

Figure 1. Block diagram of Basic AUTUMN.

Pitch extraction occurs at the next stage. AUTUMN performs pitch extraction by summing the energies of harmonically-related frequency components: if the summed energy exceeds a threshold, AUTUMN outputs a note at the fundamental frequency. Every time a note is detected, the energies in the frequency bins corresponding to harmonics of the fundamental frequency are penalised. Note determination is repeated for each window. Each window is processed independently, although there is an implicit inter-window dependency owing to the time averaging performed in the pre-detection phase.

The output of pitch extraction is then converted to a MIDI file and saved for analysis.

2.2 Time- to Frequency-Domain Conversion

AUTUMN converts the signal from the time domain to the frequency domain by using a DFT. The conversion closely follows the approach proposed by Beauchamp (1993). The block diagram presented in Figure 2 provides a conceptual breakdown of this approach, which we briefly describe here.

Windowing. A DFT is insensitive to time-variant oscillations, therefore the acoustic signal is first divided into a set of equal-sized temporal windows spanning the entire signal. The window size in AUTUMN, T_w , is indirectly specified by the analysis frequency, f_a , where the choice of f_a is discussed below. Window overlap is determined automatically according to $T_{step} = T_w/4$, which satisfies the sample-rate/bandwidth criterion (Beauchamp 1993). Each window is multiplied by a Hamming window to attenuate discontinuities at the window boundary, which would result in incorrectly strong upper harmonics. Resampling is used to

bring the number of samples in each window to the next highest power of 2, in order to exploit computational optimisations available to DFT processing.

Figure 2. Block diagram of DFT stage of AUTUMN.

DFT. Each window is now independently passed through a DFT, with analysis frequency (and therefore frequency resolution) f_a . The nature of the DFT is such that temporal and frequency resolution are inversely related according to $T_w = 2/f_a$. Any value for f_a must therefore be a compromise between frequency resolution and temporal resolution. It is not clear how to decide this compromise. Ideally, a system should be capable of resolving musically significant events, which occur at subaudio rates of $T_{sig} \geq 0.05s$ (Hall 1991). Simultaneously, we are interested in the point at which a minor third becomes resolvable. (The minor third is chosen as an indicator since it is conventionally the smallest of the most important harmonic intervals that occur in consonant music.) Minor third resolution of bass notes requires $T_w = 0.4s$.

AUTUMN defers the trade-off to the user: humans are qualified to decide on whether a performance is “slow” or “fast.” This represents a loss of autonomy in principle. In practice, experiments have shown that $f_a = 25Hz$ (therefore $T_w = 0.08s$) is an acceptable setting for most musical examples: satisfactory note resolution can be obtained with an acceptable amount of inter-event interference.

Phase Unwrapping. Following the DFT, the true location of the frequency component in each bin is known only up to the width of the bin. Using phase unwrapping, this true location can be determined more precisely. The calculated quantity is the *frequency deviation*: the distance between the centre of the bin and the true location of the component, expressed in units of frequency (e.g., Hz). This quantity is calculated from the output of the DFT.

Details of phase unwrapping are not discussed in this paper, since AUTUMN uses a straightforward phase unwrapping implementation (Beauchamp 1993).

2.3 Pre-Detection Processing

Following the process described in Section 2.2, the set of spectral snapshots are fed through a “pre-detection” stage. Two processes occur at this stage: averaging and normalisation.

Averaging. As mentioned in Section 1, it is inappropriate to average all windows when processing multi-event signals.

Nonetheless, some windows will be dominated by transient periods of attack or decay, particularly when temporal resolution is small. To mitigate this effect, AUTUMN performs “localised temporal averaging.” That is, prior to passing the spectral snapshots to the note-detection algorithm, the content of each snapshot is averaged with its nearest n temporal neighbours, where n is specified by the user. The quantity $D = n \cdot T_{step}$ is defined as the *averaging depth*, and has units of time.

The ideal value of D is not clear, and depends on both the rate of note events and the attack/decay characteristics of the instrument(s). AUTUMN leaves D as a user-specified parameter. However, empirical results show that setting $D \approx 0.1s$ typically provides a satisfying trade-off for most situations.

Normalisation. As Section 2.4 describes, a candidate note is ultimately declared as a detected note if the summed energy of all harmonics of the candidate note exceeds a threshold. It is therefore convenient to normalise the energy in each window. AUTUMN independently normalises each window according to the formula:

$$\hat{E}_i = \frac{E_i}{\sum_{j=1}^B E_j} \quad (1)$$

where E_n and $\hat{E}_n$ is the energy in bin n before and after normalisation, respectively, and B is the number of bins in the snapshot. That is, the content of each window is scaled such that the total energy of each window, summed over all harmonics, equals 1.

2.4 Note Extraction

Following the pre-detection stage, AUTUMN performs note extraction proper. The algorithm for note extraction is presented as pseudocode in Figure 5.

AUTUMN performs note-extraction on a window-by-window basis. Each window is processed independently. AUTUMN determines notes by summing the energies of harmonically related frequencies, and outputting those groups that have summed energy higher than a threshold. Upon successful note detection, AUTUMN then penalises all harmonics of that note. The process is examined step-by-step over the course of a representative sample window.

Figure 3 shows the first one hundred bins of a sample spectral snapshot, analysed at $f_a = 25Hz$. (There are 1023 bins total, however most have zero energy.) The solid horizontal line represents the average energy over all bins, in this case, $E_{avg} = 1.5\%$. Having calculated the average energy, E_{avg} , AUTUMN finds the first bin, b , that has energy $E_b \geq E_{avg}$. In this example, this is bin 12, corresponding to a frequency (after phase unwrapping) of $f_{12} = 294.9Hz$.

Frequency f_b is now treated as the fundamental frequency of a *candidate note*. AUTUMN finds all bins whose frequency peak is harmonically related to f_b , to within a tol-

erance. In other words, AUTUMN identifies the first M bins, $\{\beta_1, \dots, \beta_M\}$, that satisfy the equation:

$$f_\beta \equiv \Delta f_\beta + f'_\beta = n \cdot f_b (1 \pm \nu) \quad (2)$$

for $n = [1 \dots n_H]$

We call the set $\beta = \{\beta_1, \dots, \beta_M\}$ the *present harmonics*. The parameters ν and n_H are left as user-specifiable. The tolerance constant ν is necessary since any real instrument is not strictly harmonic. Research on instrument acoustics (e.g., Campbell and Greated 1987) suggests that a value of $\nu = 0.05$ is appropriate. The constant n_H is used to limit the number of harmonics checked, which is needed to counter the inherent low-pitch bias that results from the finite set of harmonics produced by the DFT. The algorithm is not very sensitive to changes in n_H , provided that the value is sufficiently low. (Typically $n_H = 3, 4$, or 5 in experiments.) Note that this approach is in contrast to Martins' and Ferreira's approach (2002), which terminates harmonic grouping according to the "missing harmonic" criterion. As these authors note however, the missing harmonic criterion will have difficulty detecting notes played by certain classes of instruments, e.g., instruments with missing harmonics, such as the clarinet (Campbell and Greated 1987). AUTUMN has no difficulty detecting instruments with missing harmonics, provided that the fundamental is present.

For the purposes of illustration, $\nu = 0.025$, and $n_H = 4$ in this example. AUTUMN flags bins 36 ($f_{36} = 880.1\text{Hz}$), and 48 ($f_{48} = 1180.9\text{Hz}$) as being harmonics. Bin 24 is skipped, since $\Delta f_{24} = 10.2\text{Hz} > f_{12} \cdot \nu$. The process stops at bin 48, since $n_H = 4$ bins were examined (even though only 3 bins were kept).

AUTUMN now sums the energies contained in the present harmonics, that is:

$$\tilde{E}_b = \sum_{\beta} E(\beta) \quad (3)$$

If $\tilde{E}_b \geq E_T$, f_b is declared as a *present note*, i.e., AUTUMN identifies that the fundamental frequency of the candidate note being inspected is present in the acoustic signal. Once more, E_T is left as a user-specified constant. E_T is a particularly troublesome parameter to set, since it is dependent on a number of factors, primarily on N (which is known to AUTUMN) and on the number of notes in the window (which is unknown) because of normalisation. Unlike other AUTUMN parameters, performance is sensitive even to small changes in E_T . Suitable determination of E_T is left as an open problem. In our running example, we have chosen to set $E_T = 20\%$. AUTUMN calculates $\tilde{E}_b = 18\% + 7.8\% + 2.4\% = 28.2\%$, therefore, $f_{12} = 294.9\text{Hz}$ (D_4) is output as a present note.

Upon declaration of a present note, AUTUMN penalises the set of energies $E(\beta)$ in an attempt to simulate signal subtraction. AUTUMN makes no assumption on the source instrument. Instead, it simulates a "generic instrument" by applying the following penalisation rule:

$$E_b \leftarrow 1 - \frac{1}{2} E_b \cdot \cos\left(\frac{(b-1)\pi}{B}\right) \quad (4)$$

where B is the total number of bins for the note and $b = \{1 \dots B\}$ is the bin being penalised. That is, the penalisation function is the first quarter-cycle of a cosine function. Note that only the energies of the present harmonics bins are penalised: bins containing missing harmonics and bins higher than the harmonic limit (dictated by n_H) are left untouched. In our example, the penalised spectrum is shown in Figure 4: bin 12 has been removed; bin 24 is untouched, since was not a present harmonic of f_{12} ; bins 36 and 48 have been diminished. Indeed, bin 48 is penalised to such an extent that it falls below the E_{avg} threshold and will therefore no longer trigger note detection.

Figure 3. Example spectral snapshot of an acoustic signal. Numbers correspond to frequency bin number.

Figure 4. Spectral snapshot of Figure 3 after penalty has been applied to f_{12} and harmonics.

At this point, AUTUMN proceeds to find the next bin in which the energy is greater than E_{avg} , and performs the process again. In this case, it is $f_{13} = 322.4\text{Hz}$, which will ultimately fail the note-detection process since the summed energy corresponding to harmonics of 322Hz is below E_T . The entire process repeats until the fundamental frequency of the note candidate, f_b , is $f_{B/2}$, where f_B is the highest fre-

quency output by the DFT. AUTUMN then proceeds to the next window and starts over.

The results of note-extraction are recorded in MIDI format and saved for analysis.

```

1.  LET  $E_T, v, n_H$  BE user-specified constants.
2.  LET  $B$  BE the number of bins in each window.
3.  LET  $E_b^i$  BE the Energy of bin  $b$  in Window  $i$ .
4.  FOREACH Window  $i$ 
5.    LET  $E_{avg}^i \leftarrow \sum_{m=1}^B E_m^i / B$ 
6.    FOREACH bin  $b$  in Window  $i$ 
7.       $E_b^i \leftarrow$  the energy of  $b$ 
8.      IF  $E_b^i > E_{avg}^i$ 
9.         $f_b^i \leftarrow$  the frequency that  $b$  occurs at
10.        $\tilde{E}_b^i \leftarrow 0$ 
11.        $n \leftarrow 1$ 
12.       WHILE  $n \leq n_H$ 
13.         IF  $\exists f_n^i$  s.th  $f_n^i \in [n \cdot f_b^i - v, n \cdot f_b^i + v]$ 
14.            $\tilde{E}_b^i \leftarrow \tilde{E}_b^i + E(f_n^i)$ 
15.           IF  $\tilde{E}_b^i > E_T$ 
16.             OUTPUT  $f_b^i, i$ 
17.             PENALISE( $b$ )

```

Figure 5. Pseudocode listing for note-extraction algorithm.

2.5 Summary

As Section 4 will show, Basic AUTUMN performs satisfactorily on a range of music. Nonetheless, it suffers from the drawback of having several user-specifiable parameters, representing a sacrifice of automation. (See Table 1. The “Degree of Influence” column qualitatively indicates the impact that changes in the value of the corresponding parameter have on performance.)

Table 1. List of Basic AUTUMN user-specifiable parameters.

Parameter	Degree of Influence
f_a	High
v	Low
D	Medium
E_T	Very High
n_H	Medium-Low

The parameter E_T is the most difficult parameter to automatically set, and is left as a problem for future research. For the remaining parameters, fine tuning by a human with knowledge of the signal contents can improve performance, but default values provide reasonable performance, as will be shown in Section 4. Section 3 introduces a method of eliminating or reducing the influence of two of those parameters, f_a and D .

3 V-AUTUMN

In relying on a number of user-specifiable parameters, Basic AUTUMN concedes some autonomy to

human intervention. To address this problem, we introduce Voting-AUTUMN, or *V-AUTUMN* for short. As the name suggests, V-AUTUMN attempts to reduce the dependency on parameters by implementing voting mechanisms into the transcription procedure. Specifically, V-AUTUMN makes use of two such mechanisms: *frequency voting* and *window voting*, which we discuss below. The remainder of AUTUMN, as described in Section 2, is unaffected.

Figure 6. Block Diagram for V-AUTUMN frequency voting mechanism. (The notation W_j^i signifies the i^{th} time-window analysed at frequency f_a^j .)

3.1 Frequency Voting

Section 2.2 describes the spectral/temporal resolution compromise inherent to DFT processing, which Basic AUTUMN resolves by deferring to the user. Nonetheless, Basic AUTUMN is incapable of simultaneously resolving rapidly-occurring events and low-pitched notes. V-AUTUMN achieves both these effects by performing multiple DFTs, each having a different f_a . Figure 6 illustrates this schematically. In particular, the original acoustic signal is copied n times, and each copy is independently processed according to the Basic AUTUMN formulation, with the DFT analysis frequency of the j^{th} copy being set to f_a^j . However, prior to MIDI output, the n copies are recombined on a window-by-window basis by a voting mechanism. The voting mechanism itself is straightforward: for each window, V-AUTUMN examines the note-extraction output of each copy, and every time a note appears, it is given a vote of ‘+1’. Once all the copies have been analysed for a given window, all notes that have received $\lfloor n/2 \rfloor + 1$ votes (i.e., a fifty-percent-plus-one majority) are output as *Voted Detected Notes*, and are passed on for further processing. Notes with less than this majority threshold are discarded. In this way, V-AUTUMN exploits the high resolution of “extreme” settings of f_a , while simultaneously benefiting

from the generally accurate mid-range values of f_a . In addition, less human intervention is required by eliminating f_a as a parameter.

Table 2. Summary of f_a^j used in V-AUTUMN

f_a^j [Hz]	T_w^j [ms]	Minimum Minor Third Resolution [Note/Note (Hz/Hz)]
10	200.0	G# ₁ / B ₁ (52 / 62)
14	142.9	D# ₂ / F# ₂ (78 / 92)
25	80.0	C ₃ / Eb ₃ (131 / 156)
35	57.1	F# ₃ / A ₃ (185 / 220)
45	44.4	B ₃ / D ₄ (247 / 294)
55	36.4	D ₄ / F ₄ (294 / 349)
63	31.7	F ₄ / Ab ₄ (349 / 415)

V-AUTUMN votes over seven signal copies. The f_a^j are summarised in Table 2. Note that the f_a^j are pair-wise relatively prime. This is intentional to maximise information gain. Note also that the best temporal resolution provided for by V-AUTUMN is $T_w^7 = 31.7$ ms. In particular, it is sufficiently sensitive to capture the fastest musically significant events.

An undesired side-effect of this frequency voting mechanism is that the effective post-voting window size is dramatically increased, since V-AUTUMN has an output window rate of:

$$T_{output} = \frac{1}{2f_{output}} \quad (5)$$

where $f_{output} = \text{LCM}(f_a^j)$. In the case of V-AUTUMN,

$T_{output} \approx 14\mu\text{s}$, roughly a thousand times faster than musically significant events, and consequently needlessly perceptually unstable. V-AUTUMN's strategy for dealing with this is discussed in Section 3.2.

3.2 Window Voting

V-AUTUMN exploits the additional information generated as a result of frequency to ultimately bolster perceptual stability by adding a second layer of voting: *Window Voting*. This is depicted schematically in Figure 7. In adding Window Voting, the Window Averaging mechanism in the Pre-Detection Processing stage (Section 2.3) is dropped.

Window voting is conceptually similar to the window averaging proposed in Section 2 in that, in both cases, the information contained in several neighbouring temporal windows is combined. However, they differ in two key aspects.

Firstly, window voting operates over detected, frequency-voted notes, whereas window averaging in Basic AUTUMN operates on the raw spectral snapshots. The result of the Basic AUTUMN paradigm is that perceptual stability is achieved through a type of spectral “blurring.” Instead, V-AUTUMN's approach establishes robustness

through “consensus.” The difference is dramatised by considering the toy example illustrated in Figure 8. Figure 8a contains the acoustic signal of a sinusoidal A₃ followed by a sinusoidal C₅. The figure also contains the location of the windows. Figure 8b illustrates the spectral snapshots corresponding to the two windows. Let $E_T = 0.7$, and for the purposes of this simple example, let a vote-count of 50% (not 50% + 1) be sufficient for a successful voted detected note. Basic AUTUMN averages the spectral information of the two windows (Figure 8c) and passes the result to the Note Detector – which will fail both notes, since the energy of each note is less than 0.7. Conversely, V-AUTUMN will pass the spectra of Figure 8b as-is to the Note Detector. The Note Detector will output one note in each window, after which the Window Voting mechanism will produce one window with both notes. The two mechanisms are therefore not equivalent.

Figure 7. Block diagram of Window Voting mechanism in V-AUTUMN.

Secondly, the two methods differ in the way transience is approached. Window Averaging can be viewed as an “additive mechanism”: transient events (e.g., attack, decay) are drowned out by adding steady-state information from neighbouring windows. On the other hand, window voting is a “subtractive mechanism,” by which transient features are effectively removed by the dominance of temporally-neighbouring steady-state information. Viewed another way, Window Averaging may be roughly viewed as a type of greedy mechanism, whereas Window Voting is conservative. This is significant, since false hits (notes output by the Note Detector but that are not present in the original signal) are perceptually more disturbing than false misses (notes in the original signal that are not detected by the Note Detector.)

In V-AUTUMN, the size of the voting population is user-specifiable but a natural default setting is 50ms, i.e., the speed of musically significant events.

4 Results

The output of V-AUTUMN was evaluated using extensive listening tests, consisting of twenty-six test subjects and forty-three musical selections. The selections each had a duration of roughly ten seconds (one or two musical

phrases). V-AUTUMN was made to automatically transcribe each of the selections into MIDI files based solely on the original acoustic signal. (Basic AUTUMN transcriptions were not performed, since V-AUTUMN perceptually outperforms Basic AUTUMN.) Independently, each of the files was hand-transcribed to MIDI, using the same instrumentation as AUTUMN's output (MIDI instrument patch 54). Each test subject listened to all forty-three selections. For each selection, the subject first listened to the human transcription, followed by the AUTUMN transcription, then rated the AUTUMN transcription according to three categories: *Overall Similarity* (OS), *Melodic Recognisability* (MR), and *Degradation* (DEG). Each ranking was made on a scale of 1 to 9, with 9 being the best and 1 being the worst.

Figure 8. Example illustrating the consequences of Window Averaging. (a) a sinusoidal A_3 (220Hz) followed by a sinusoidal C_5 (520Hz) (windows shown); (b) spectral snapshot corresponding to each window; (c) spectral snapshot after Window Averaging.

Results are summarised in Table 3, divided according to instrumentation. In all cases, tests subjects rated V-AUTUMN's output as having a roughly "average similarity" in all three categories. Not surprisingly, V-AUTUMN performs best on signals with few instruments: thin instru-

mentation facilitates note identification since harmonic relationships are obvious in the spectrum. AUTUMN also performs perceptually well on signals with one dominant instrument. This is also as expected, since the harmonics of the dominant instrument will dominate the spectrum. Note that melodic recognisability was ranked consistently higher than overall similarity and degradation across all instrumentation categories. This is encouraging, since transcription applications are typically most interested in the melodic line.

Table 3. Summary of results of listening tests. "OS" is "Overall Similarity," "MR" is "Melodic Recognisability," and "DEG" is "Degradation." All tests are ranked on a scale of 1 to 9 (worst to best, respectively).

Instrumentation	OS	MR	DEG
1-3 instruments	5.7	6.1	5.5
4-6 instruments	4.2	4.7	4.4
>6 instruments, 1 dominant	4.6	4.8	4.6
>6 instruments, several dominant	4.1	4.4	4.1
Vocal/Operatic	3.9	4.3	3.9
Pop	4.1	4.3	3.9
Overall	4.4	4.7	4.3

Not surprisingly, perceptual performance decreases as the number of instruments increases: as more notes – and therefore harmonics – are added, spectral distribution becomes blurred, thus making the correct harmonic grouping ambiguous. Performance was also decreased in the presence of vocals and more modern instruments, both of which make use of some degree of musical noise. Operatic music was additionally troublesome: although the emphasis on vowel sounds was helpful to AUTUMN, the high degree of vibrato characteristic of operatic singing in particular often results a semitone instability in the AUTUMN output. This last failure, however, is a shortcoming in the expressiveness of MIDI. Judicial use of the MIDI pitch bending feature might improve perceptual output.

In addition to listening tests, some quantitative analysis was performed on Basic and V-AUTUMN transcriptions. Both true hits (notes in both the original and the transcription) and false hits (notes in the transcription that were not in the original) were counted. Owing to the difficulty associated with establishing ground-truth knowledge of the original signal (e.g., precise timing of note-events), this analysis was performed only over three representative signals.

The first test signal was a recording of a synthesized solo clarinet playing a portion of G. Mahler's 4th Symphony (Horner and Ayers 2002). AUTUMN was presented with only the acoustic data: all symbolic data used to generate the performance were unknown to AUTUMN. The choice of clarinet is significant because of the missing even harmonics characteristic of the instrument. In this case, most settings of Basic AUTUMN could achieve over 90% true hits, with false hit rates of under 10%. The best setting of Basic AUTUMN achieves 97% true hits, with 5% false hits. V-

AUTUMN detects 95% of all notes, with a false hit rate of 0.5%. If false octave hits (false hits which are exact multiples of a note present in the original signal) are factored out, V-AUTUMN makes exactly zero incorrect detections. (False octave hits are perceptually special, since the human ear is more forgiving towards notes in an octave relation.)

The second test consisted of the same composition as above, but with the complete original instrumentation – two synthetic clarinets and two synthetic horns. On average, Basic AUTUMN achieved a true-hit detection rate of roughly 70%. False hits rose significantly, to between 15% and 75% – that is, the noisiest settings of Basic AUTUMN detected more incorrect notes than correct ones. The best-tuned settings resulted in true- and false-hit rates of 78% and 13%, respectively. On the same example, V-AUTUMN detected 71% of all notes, while limiting the false hits to 3.6%.

Lastly, the test was performed using a commercial (non-synthesised) recording of a Bach cantata, arranged for brass quartet. Basic AUTUMN true hit detection remained at roughly 65% (71% for the best settings), but false hits dominated the output at roughly 100%, regardless of the parameter settings. In other words, Basic AUTUMN detected as many false notes as notes present in the original signal. In contrast, V-AUTUMN maintained a true hit rate of 67%, while limiting false hits to 20% (and only 3% if false octave hits are factored out).

These results suggest that voting mechanisms make V-AUTUMN extremely robust to noise, correctly filtering out incorrect notes even in transcriptions where incorrect notes dominate correct ones. Significantly, this filtering is done with minimal loss of correct detections: indeed on the three test cases, V-AUTUMN fell only slightly short of the best Basic AUTUMN settings, and always outperformed the average case. Additionally, a comparison of the quantitative results with the listening test scores suggests that AUTUMN's primary weakness is a lack of stability: while it detects the correct notes in the majority of cases, the detection is not sustained over the entire duration of the window (since detection is performed on a window-by-window basis).

5 Conclusions

This paper has introduced basic AUTUMN, an automatic music transcription system that extracts pitches present in an acoustic signal on a window-by-window basis. AUTUMN was designed to perform on signals of arbitrary content: the signal is permitted to have any number of instruments, any type of instrument, and any number of notes. Human intervention is still required, but reduced through the voting mechanisms incorporated into V-AUTUMN. Reducing human intervention further is a direction of future research.

Results of listening tests suggest that AUTUMN performs modestly well, but there remains room for improvement. Preliminary comparisons of perceptual results with

quantitative analysis of AUTUMN output suggest that AUTUMN suffers most from a lack of stability. Future work will focus on means of providing this stability. Additionally, the heuristic penalisation mechanism may be responsible for a large proportion of false octave and false harmonic hits. Markovian processing, less naïve voting, and musically-knowledgeable systems are conjectured to have a positive effect on performance.

6 Acknowledgements

This work was supported in part by the Hong Kong Research Grant Council's Projects HKUST6167/03E and HKUST6135/05E.

References

- Beauchamp, J. 1993. Unix Workstation Software for Analysis, Graphics, Modification, and Synthesis of Musical Sounds. *Audio Engineering Society Preprint* (No. 3541). Berlin: AES.
- Campbell, M., and Greated, C. 1987. *The Musician's Guide to Acoustics*. New York: Schirmer Books.
- Goto, M. 2001. A Predominant-F0 Estimation Method for CD Recordings: MAP Estimation Using EM Algorithm for Adaptive Tone Models. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*. pp. 3365-3368. Salt Lake City: IEEE.
- Hall, D.E. 1991. *Musical Acoustics* (Second Edition). Pacific Grove: Brooks/Cole Publishing Company.
- Hainsworth, S.W., and Macleod, M.D. 2001. Automatic Bass Line Transcription from Polyphonic Music. In *Proceedings of the International Computer Music Conference 2001*. Havana: International Computer Music Association.
- Horner, A., and Ayers, L. 2002. *Cooking with CSound*. Middleton: A-R Editions.
- Klapuri, A. 1999. Pitch Estimation Using Multiple Independent Time-Frequency Windows. In *Proceedings 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. pp. 115-118. New Paltz: IEEE.
- Klapuri, A., Virtanen, T., Eronen, A., and Seppanen, J. 2001. Automatic Transcription of Musical Recordings. *Consistent and Reliable Acoustic Cues Workshop*. Aalborg: CRAC-01.
- Klapuri, A. 2003. Automatic Transcription of Music. In *Proceedings of the Stockholm Music Acoustic Conference*. pp. 1-3. Stockholm: SMAC.
- Martins, L.G.P.M., and Ferreira, A.J.S. 2002. PCM to MIDI Transposition. In *Proceedings of the 112th Convention of the Audio Engineering Society*. Munich: Audio Engineering Society.
- Monti, G., and Sandler, M. 2002. Automatic Polyphonic Piano Note Extraction Using Fuzzy Logic in a Blackboard System. In *Proceedings of the 5th International Conference on Digital Audio Effects*. pp. 39-44. Hamburg: DAFx.
- Sieger, N.J., and Twefik, A.H. 1997. Audio Coding for Conversion to MIDI. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing*. pp 101-106. Leicester: IEEE.
- Tolonen, T., and Karjalainen, M. 2000. A Computationally Efficient Multipitch Analysis Model. *IEEE Transactions on Speech and Audio Processing* 8(6). pp. 708-716. Washington, D.C.: IEEE.