EXTRACTING MELODY LINES FROM COMPLEX AUDIO

Jana Eggink & Guy J. Brown

Department of Computer Science, University of Sheffield, UK
{j.eggink, g.brown}@dcs.shef.ac.uk

Abstract

We propose a system which extracts the melody line played by a solo instrument from complex audio. At every time frame multiple fundamental frequency (F0) hypotheses are generated, and later processing uses various knowledge sources to choose the most likely succession of F0s. Knowledge sources include an instrument recognition module and temporal knowledge about tone durations and interval transitions, which are integrated in a probabilistic search. The proposed system improved the number of frames with correct F0 estimates by 14% compared to a baseline system which simply uses the strongest F0 at every point in time. The number of spurious tones was reduced to nearly a third compared to the baseline system, resulting in significantly smoother melody lines.

1. Introduction

This paper is concerned with obtaining an adequate symbolic representation from musical audio. The word 'adequate' indicates that we are not trying to extract all possible information, but only information which is relevant for a specific task. All information would include a full transcription in form of a musical score, detailed acoustic analysis of the instruments playing, emotional expression, harmonic analysis and so on. This is not only a currently infeasible task, but would also result in a large amount of what is likely to be irrelevant information for a specific task. Here, we focus on the extraction of the major melody line from polyphonic audio, which we define as the melody played by the solo instrument in an accompanied sonata or concerto. While this definition is not necessarily always accurate in terms of perception or music theory (since, for example, another instrument from the orchestra could play a short solo), it provides a good evaluation basis. It is also likely to be an adequate definition for most practical applications. Applications for which an automatic melody extraction would be useful include automatic indexing and analysis, detection of copyright infringement, and 'query-by-humming' systems. In particular, the latter has attracted increasing attention over the last few years (e.g., see [8] and other publications from the same conference). But so far, these systems rely on manual coding of the melody line, which is then used to match against the melody 'hummed' by the user. The system introduced in this paper attempts to bridge this gap by providing a melody extracted directly from audio, without the need for manual interference.

Previous work has tended to concentrate on solving the problem of obtaining a full transcription, i.e. extracting all fundamental frequencies (F0s) from polyphonic music. But no general solution has been proposed so far; the musical material has always been restricted in terms of the maximum number of concurrent notes, specific instruments and musical style. Some examples include the work of Klapuri [7], who concentrated on rich polyphonic sounds with relatively high numbers of simultaneous notes, but only used mixtures of isolated tone samples. Working in the frequency domain using a subtraction scheme based on spectral smoothness he obtained accuracies between 54% and 82% for 6 concurrent F0s. Raphael [9] pursued another approach, which was restricted to piano music by Mozart with no more than 4 concurrent tones, but employed naturalistic recordings. His system is based on a statistical modelling approach using hidden Markov-models (HMMs), and he achieved an accuracy of 61%.

Goto [6] pursued an approach in many ways related to the one proposed in this paper. Instead of trying to estimate all F0s, he concentrated on estimating only the F0s of the melody and the bass line in limited frequency ranges. Using adaptive tone models within a multi-agent architecture, he obtained good results of around 88% correctly identified F0s on a frame-by-frame basis for the higher pitched melody, and slightly less for the bass line. His system was mainly evaluated on popular and jazz music with the melody line produced by a singer, and no attempt was made to distinguish between sections where the singer is present or silent.

Berenzweig [2] investigated the distinction between sections of a piece of music where the singing voice is present or absent. He compared statistical models trained directly on acoustic features or on the likelihood outputs of a speech recogniser trained on normal speech. He found little advantage of the latter, suggesting that the differences of sung words with background music from normal speech are too large to be useful; and no attempt was made to separate the voice from the background accompaniment. He obtained correct classifications in 74% of frames, improving to 81% when averaged over 81 frames.

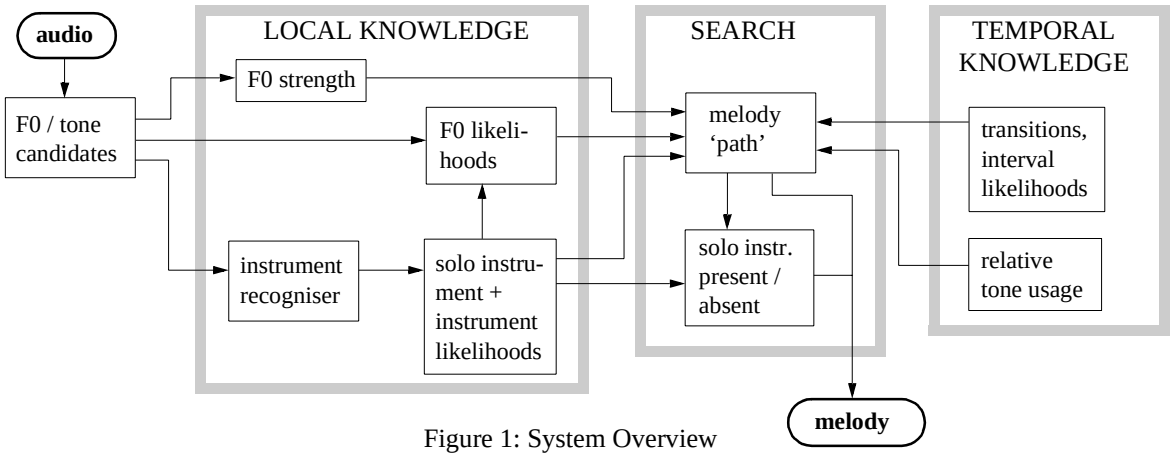


Figure 1: System Overview

For the current system we assume a classical sonata or concerto, where a solo instrument is accompanied by either a keyboard instrument or an orchestra. Solo instruments can span the whole pitch range, ranging from the low tones of a cello to a high pitched flute, so that a division of the frequency range in the way Goto proposed for his system [6] is not feasible. It is also not guaranteed that the solo instrument will always produce the strongest F0, so that additional knowledge is necessary to extract the melody line. One of the major sources of information – and a potential advantage of our system compared to existing ones – is the integration of an instrument recognition module. This knowledge can both help to choose the correct F0 among multiple concurrent F0 candidates, and to determine sections where the solo instrument is actually present as opposed to sections where only the accompanying instruments are playing.

2. System Overview

The main objective of the present system is to provide a framework for integrating different knowledge sources, in order to enable the robust extraction of the main melody. Finding the melody is here defined as finding the succession of F0s produced by the solo instrument. If the strongest F0 at every time frame was always played by the solo instrument, then all that would be needed is an algorithm to find this F0. The strongest F0 might be defined as the perceptually most salient one, although that still leaves open the question for a definition in acoustic or signal processing terms. Reasoning from a musical point of view, the solo instrument is more likely to have the strongest F0 in concertos since, for example, the solo violin is still perceivable as such even when accompanied by a whole set of first and second violins within the orchestra. This seems less obvious in a sonata with piano accompaniment, where even the title might indicate an equal importance of both instruments by changing the order in which the instruments are named, for example sonatas for piano and violin by Beethoven or Brahms. Instead of trying to find only the single most prominent F0 at every point in time, we therefore extract multiple F0 candidates and use additional knowledge at later processing steps to decide which of the candidates belongs to the melody played by the solo instrument.

The knowledge sources can be divided into two categories, local or stationary knowledge and temporal or transitionary knowledge. Local knowledge includes the strength of every F0 hypothesis and the likelihood of that F0 to be produced by the solo instrument, both in terms of instrument recognition, and in terms of the pitch range of the solo instrument. Temporal knowledge includes measures related to tone durations and intervals between successive tones. This knowledge is combined to find the most likely 'path' through the space of all F0 candidates in time; in other words, to find the melody. As the melody path occasionally follows the accompaniment, additional 'post processing' is carried out to eliminate sections where the solo instrument is actually silent.

In the following we will assume that the identity of the solo instrument is known. We were able to classify the solo instrument correctly with high accuracies of over 85% in an earlier system [5]. For now we assume that this instrument identification has been carried out in advance, although a more integrated solution is of course preferable, and will be addressed in further research.

2.1. F0 and Tone Candidates

2.1.1. Time-Frequency Conversion

The first processing step is a conversion from the time into the frequency domain. Most commercially available music is recorded at 44100 Hz, and this sampling frequency is retained. The signal is divided into frames of 70 ms (3072 samples) length with 33% (1024 samples) overlap. Every frame is multiplied with a Hanning window and a fast Fourier transform (FFT, size 16384) is computed. The spectral power is log-compressed for all further processing. The window size is relatively large compared to most audio processing applications, but proved to be necessary to ensure a sufficient resolution for low frequencies. Other research [7] used even larger windows of up to 190 ms for F0 estimation in polyphonic music, and experienced a strong decrease in accuracy for shorter windows lengths.

2.1.2. Peak Extraction

The next processing step is concerned with data reduction. Instead of using the complete spectrum for further processing, only spectral peaks are kept. These are used both to estimate F0s, and as a basis for instrument recognition. To estimate peaks the spectrum is convolved with a 50 samples wide differentiated Gaussian. As a result the spectrum is smoothed and peaks are transformed into zero crossings, which are easy to detect. The frequency of a peak is then defined by its corresponding FFT bin. A highly zero-padded FFT is used, in order to increase the accuracy of the frequency estimates for the spectral peaks. Spectral peaks are discarded if their power is below a threshold. This threshold differs with frequency to account for the fact that musical instrument tones generally have more energy in lower frequencies. We employ a local threshold defined by the mean of the spectral power within a range of $\pm$ 500 Hz around a peak.

2.1.3. Estimating F0 Candidates

Different methods of F0 estimation were investigated, with the first one searching for prominent harmonic series within the spectral peaks of a frame. While this led to good results if two instruments were playing at approximately the same level ([4]), it worked less well in the present study. One specific problem is that the accompaniment often plays F0s one or two octaves below the solo instrument, which regularly resulted in the solo instrument being estimated one or two octaves below its true F0 value. Comparing the search for a prominent harmonic series with an F0 estimation that is based solely on the power of isolated spectral peaks we found no advantage of the former for most solo instruments. Since our approach depends on the existence of a spectral peak at the fundamental frequency, it cannot account for perceptual phenomena such as the 'pitch of the missing fundamental', but proved suitable for the musical instrument tones used in the present study. Using the strongest spectral peaks as F0 candidates also has the advantage that it has low computational cost.

2.1.4. Connecting Peaks to Form Tones

To correct isolated errors made by the F0 estimation algorithm, and to provide a more restricted basis for the search module, frame-based F0 hypotheses are connected to form longer tones. The approach is based on a birth-and-death algorithm, where tones continue if there is a matching F0 hypothesis in the next frame, and are terminated otherwise, while unmatched F0s hypotheses start a new tone. The matching interval for the continuation of a tone is based on its average frequency up to that point. The use of an average frequency seems preferable over an estimated trajectory, as it allows for vibrato while breaking up successive tones even when they are separated by only a small interval. To account for potential errors or misses in the F0 estimation algorithm, tones can continue even if for one frame (or maximally two frames) no matching F0 hypotheses are found. In these gaps the exact locations and strengths of missing F0s are reestimated by searching the spectral peaks for a candidate that is within the matching interval of the average F0 of the tone. A minimum tone length of two frames is imposed, which helps to suppress isolated random errors in the generation of F0 hypothesis. In a second processing step, tones are split when sudden discontinuities in F0 strength occur. This allows a new tone to be started if, for example, the solo instrument starts on the same F0 as a long note played by the accompaniment.

2.2. Local Knowledge

All local knowledge is derived from the frame-based F0 candidates. Every candidate has three attributes, which describe its spectral power, its fundamental frequency, and a set of features derived from its harmonic overtone series used for instrument recognition.

2.2.1. F0 Strength

Assuming that the solo instrument is commonly played louder than (or at least as loud as) the accompaniment, and that strong F0 estimates are less likely to be caused by artefacts, the likelihood of an F0 candidate can be directly inferred from its spectral power. The stronger the spectral peak that caused the F0 hypothesis, the higher its likelihood.

2.2.2. F0 Likelihood

If the solo instrument is known, the likelihood of an F0 candidate in terms of its frequency can be estimated. In a first step, all candidates with F0s outside of the range of the solo instrument are discarded. Additionally, even within the range of an instrument not all F0s are equally likely; for example, the F0s at the end of the range of an instrument are normally less common. We computed the likelihood of an F0 for a specific instrument by counting the frequency of its occurrence in MIDI (Musical Instrument Digital Interface) transcriptions of 4-5 different sonatas and concertos. To avoid influence by the keys in which the pieces were written, likelihood values were smoothed across frequencies.

2.2.3. Instrument Likelihood

Instrument recognition is carried out for every frame-based F0 estimate of every tone. This part of the system was developed earlier to enable recognition of solo instruments without imposing any restrictions on the background; for details see [5]. It uses a representation based solely on the F0 and partials of an instrument tone and has been shown to be highly robust against background accompaniment. The features consist of the exact frequency location and the normalised and log-compressed power of the first 10 partials of the target tone. Added to these are frame-to-frame differences (deltas) and differences of differences (delta-deltas) within tones of continuous F0. A Gaussian classifier with a full covariance matrix is trained for every possible F0 of

every instrument; the classifier which maximises the posterior probability of a set of test features determines its instrument class.

Instead of making hard decisions and assigning every tone to an instrument class, the likelihood values of it being produced by the solo instrument are computed and later used as one knowledge source among others in the melody search.

2.3. Temporal Knowledge

2.3.1. Interval Likelihood

Different interval transitions between successive tones within a melody have different likelihoods; in western classical music small intervals are generally more common than large ones. Similar to the instrument dependent F0 likelihood, we computed instrument dependent interval likelihoods from 4-5 different MIDI files. The distributions were close to those expected from common musical knowledge, with a full tone being the most frequent interval, and nearly monotonically decreasing likelihoods for larger intervals, slightly favouring fifths and octaves. Instrument-dependent variations were very small and might have been caused primarily by the selection of examples, but were kept for the current evaluation.

Intervals are computed between the average F0s of successive tones. The likelihood value for every interval is multiplied with the length of the second tone, since otherwise two or more short tones with relatively likely interval transitions would be favoured over one long tone.

2.3.2. Relative Tone Usage

The measure of relative tone usage is solely used as a punishment under special conditions, where the path includes the start of a tone but leaves it before the tone ends. This is necessary to allow for reverberant conditions which can cause an overlap between successive tones. To discourage paths that leave a tone very early and while the tone has still strong F0 candidates, the percentage of frames unused within that particular tone is calculated and weighted by the mean F0 strength of these skipped F0 hypotheses. If all that is skipped is a reverberant 'tail', the number of frames will be limited and will have a relatively low F0 strength, so that the punishment will be small. In all other cases the path is probably trying to leave a tone that really continues and therefore the path's overall probability is correctly diminished.

2.4. Search

2.4.1. Finding the Melody 'Path'

The different knowledge sources outlined in the previous section are combined to find the most likely 'path' through the maze of all possible F0s candidates over time. This resembles in many ways a tracking problem, such as one in which the changing location of a moving person has to be identified. A common way of knowledge integration for this type of problem is the Bayesian framework, for which a number of computationally efficient algorithms exist to find the optimal path (e.g. [1]). One of the preconditions for most of these algorithms is that the Markov assumption holds true, e.g. the likelihood for a state depends on the previous state, but not on earlier or future states. This is not the case within our system, where the temporal knowledge sources violate this precondition. Also, the computational complexity in our application is strongly constrained by the fact that not all positions within the space of all possible F0s are considered at every time frame, but only those for which an F0 hypothesis exists. Nevertheless, evaluating all possible paths is far too costly, and we therefore apply some pruning. A simple *n*-best method is used, where at every time step only the best *n* paths are kept.

The final melody path contains one F0 for every frame, but the search for this path is restricted and guided on the basis of longer tones. The rules for allowed movements within the time-F0 space are mainly derived from general musical knowledge and are as follows:

- Every tone within the melody path has to be included from the tones' beginning.
- A tone has to be followed to its end, unless a new tone starts.
- A return to the same tone is prohibited once the path has left this tone.
- After a tone has finished, the path can either go to a new tone or to a silence state.

Within all possible paths that obey these restrictions, the knowledge sources help to find the most likely melody path. The overall likelihood of a path at a certain point in time consists of the sum of both the local and the temporal knowledge sources up to that point in time. The knowledge sources are weighted with different importancy factors ('weights'), to reflect the fact that different types of knowledge might be of different importance for an optimal path finding solution.

To enable comparisons across different sources of information, all probabilities are normalised to have zero mean and a standard deviation of 1. Specifically, this normalisation is carried out for each sound file across all instrument likelihood and F0 strength values. Additional knowledge sources have fixed values independent of the actual example, but are set within a similar range of values. During frames which are labelled as silent all knowledge sources are set to their mean value of 0.

2.4.2. Silence Estimation for the Solo Instrument

The solo instrument in classical sonatas or concertos is not necessarily playing continuously. The melody path finding algorithm is able to label a frame as silent if none of the available tones leads to a higher confidence value. But at the current stage, this is not reliable enough to automatically detect longer sections where the solo instrument is silent, where the path frequently picks up F0 candidates belonging to the accompaniment. We therefore

developed an additional processing step to determine the state of the solo instrument being 'present' or 'silent' at different moments in time.

The melody as estimated by the path finding algorithm with the corresponding likelihoods for the solo instrument is used to distinguish between these two states, assuming that this likelihood will be higher for sections where the solo instrument is present than for those where it is silent. To reduce the influence of random variations and outliers in the likelihood estimates from the instrument recogniser, median smoothing is applied across neighbouring likelihood values along the melody path. Two thresholds are used to estimate sections where we can be fairly sure about the state of the solo instrument. The present threshold is set at the median of all solo instrument likelihoods, starting from the assumption that the solo instrument is present at least half the time. The silent threshold is set to zero, the mean of the likelihoods over all instruments and all F0s. Should the present threshold be less than the silent threshold, the silent threshold is lowered until it becomes the lower one. This scenario should not happen if the other parts of the system are working well; it might be an indication that the overall instrument recognition scheme has produced a false estimate for the solo instrument.

After these two thresholds are defined, every tone is investigated and if at least 75% of its frames are within the definite present or silent range, the whole tone is assigned to that state. In a second processing step, unassigned tones are investigated. If the previous and the following tone of a candidate tone are both assigned to the silent state, the candidate tone in the middle is assigned to the same state, as isolated tones by the solo instrument are unlikely. Remaining tones are assigned according to the median of their instrument likelihood values. If this is closer to the present threshold, the tone is assigned to this state, otherwise it is assigned to the silent state. Additionally, it is assumed that very short sections of presence for the solo instrument are unlikely. A minimal duration threshold is applied, which declares sections where the solo instrument is declared present as spurious if their length falls below that threshold.

3. Evaluation

The major part of the evaluation is based on audio files generated from MIDI data. This allows for a direct control of the stimuli and provides a good basis for automatic evaluation procedures. 5 different solo instruments were used, flute, clarinet, oboe, violin and cello. The main reason for this choice was that these are commonly used as solo instruments, and unaccompanied CD-recordings for training purposes were locally available. Both concertos with orchestral accompaniment and sonatas with keyboard (piano or cembalo) accompaniment were used for evaluation; examples were chosen so that every solo instrument was accompanied by a each type of accompaniment. From each of the 10 examples either a whole movement was taken, or, in the case of very long examples, the first 3 minutes. No specific consideration was given to the tempo of the pieces, but both pieces with fast and slow musical tempi (e.g. allegro, adagio) were used. No selection according to specific musical style was performed; while all pieces can be categorized as western classical music, they span a range from the baroque (late 17th to early 18th century) to the romantic period (early to late 19th century). All MIDI files were taken without further editing from the classical music archives [3]. They were played using samples provided by Propellerhead's Reason 2.5 sampler software, which in most cases used 4 recorded tones per octave. The solo voice and the accompaniment were first saved in independent sound files and then mixed at 0 dB root-mean-square (rms) energy levels. Sections where the solo instrument is silent were ignored for the normalisation procedure, as otherwise the solo instrument would be stronger compared to the accompaniment in pieces where it is not playing continuously. The choice of mixing levels is somewhat arbitrary, and of course, the stronger the solo instrument, the easier the task of estimating the corresponding F0s. Informal listening tests by the authors showed that a mixing level of 0 dB was acceptable as a realistic option, with the solo instrument being identifiable as such.

The frequencies provided by the MIDI files were taken as ground truth. In some pieces for violin or cello the solo instrument is in fact playing more than one F0 simultaneously; in these cases either one would be considered correct. Evaluation was carried out both in terms of the percentage of correctly estimated F0s on a frame-by-frame basis, and based on longer tones of continuous F0. No explicit onset detection was carried out, so that repeated tones without silence in between were automatically merged into one longer tone. A tone was considered as 'found' if at least one frame of it was matched by the estimated melody path, while all tones in the estimated melody which did not at least partially match with the true F0s were taken to be spurious tones.

Automatic evaluation of realistic recordings taken from commercially available CDs is less straight forward. We manually annotated two short examples with the correct F0s using a mixture of listening and visual spectrogram inspection to place the notes taken from a score at the correct time frames. These two examples are mainly considered as proof of concept, as the test set is too limited for a full quantitative evaluation.

3.1. F0 Estimation

All further processing relies on the F0 and tone candidates estimated in the initial processing step. Tones missed can not be recovered at a later stage, so additional false candidates seem preferable over missed ones. Using simply the one strongest spectral peak as F0 candidate, this F0 was correct for 52% of frames on average (based only on those frames where the solo instrument is actually present). Increasing the number of F0 candidates

improved the results, counting as correct every frame where any one of the F0 candidates matches the true F0. With 3 candidates correct F0s were found in 76% of frames, increasing to a plateau of around 95% with 15 or more F0 candidates. F0 estimation was better for woodwinds than for strings, which is probably at least partially due to the fact that the strongest partial coincides more often with the F0 for woodwinds than for strings. Differences between the instruments became smaller with more F0 candidates, see Table 1. Additionally, F0 estimation was better for concertos than for keyboard accompaniment for all instruments, with an average difference of 10% with both 1 and 3 F0 candidates.

F0 candidates	flute	clar.	oboe	violin	cello	average
1 (strongest)	70%	78%	48%	28%	38%	52%
3	82%	94%	78%	61%	64%	76%
15	98%	99%	98%	94%	84%	95%

Table 1: Frames with correct F0 estimates (based only on sections where the solo instrument is present), with the number of F0 candidates in rows and the different instruments in columns.

3.2. Instrument Recognition

The instrument recognition scheme has been previously shown to give good recognition accuracies, with no drop in performance between monophonic and accompanied phrases [5]. Tested on the melodies without accompaniment and using knowledge about the correct F0s, all examples were correctly classified, except for one piece for oboe, where the instrument was mistaken for a flute. However, recognition performance in the presence of accompaniment was, even when evaluated based on correct F0s only, well below expectations. While violin and cello were still correctly identified by the system, the 3 woodwind instruments were repeatedly confused with each other or a string instrument.

These results indicate that either samples are harder to identify than realistic phrases, maybe because they show less of instrument typical variations such as vibrato, and recognition performance is therefore less robust in the presence of interfering sound sources. The other factor likely to have a strong influence is the mixing level between solo instrument and accompaniment; and the frequency regions dominated by the accompaniment. The instrument recognition results could suggest that the mixing used results in a SNR that is actually less favourable for the solo instrument than those found in professional recordings of real musical performances.

3.3. Melody Path

All knowledge sources used to find the most likely path are weighted by individual importancy factors. The setting of these weights influences the performance of the system to a large degree. The weights were estimated in a hill climbing procedure, where the weights of every knowledge source were randomly increased or decreased, to find new weight combinations which led to a better estimate of the known melody path. Comparing the best knowledge weights estimated for the different examples, some similarities could be observed. The strength of the F0 was in all examples one of the most important factors, often with a weight more than double that of the next important knowledge source. The tone usage factors was not useful in more than one or two examples, but this could be partly due to the data, as overlapping tones or tones with inaccurately estimated onsets are less common in MIDI based audio than in realistic recordings. Knowledge about the instrument dependent likelihood of a specific F0s was useful in almost all cases, as was the likelihood of a F0 to be produced by the solo instrument. The latter was true even for examples were the overall instrument recognition was poor. This suggests that the likelihood of having been produced by the solo instrument was still higher for the true F0s than for spurious ones or those belonging to the accompaniment.

For further evaluation the combination of weights that gave the best results independent of the specific example was used, with an importancy factor of 7 for the F0 strength, 2 for the F0 likelihood, 2 for the instrument likelihood, 3 for the interval likelihood, and 0 for the relative tone usage. Adjusting the weights individually gave only a relatively small improvement of around 5% more correctly identified frames, which suggest that a stable parameter set can be obtained which is independent of particular musical examples.

Another factor influencing performance is the number of F0 candidates considered, which is closely connected to the amount of pruning used in the search for the best path. To keep computational costs within a reasonable amount, we used only 3 F0 candidates per frame. If the path finding algorithm was able to always choose the correct F0s, this allows for an increase of 25% overall correct frames over the baseline of choosing the strongest F0s at every frame. Again due to necessary constraints in computation costs, we limited the search algorithm to a maximum of 5 paths. Spot checks using different knowledge weight combinations showed that less pruning improved results only for a very limited number of examples, and even then only marginally.

Simply taking the strongest F0 at every time frame can be seen as baseline performance, since no knowledge is involved. As with the path finding at this stage, no attempt is made to determine sections where the solo instrument is present or absent, and we therefore only use sections where the solo instrument is present for evaluation. Using the same set of weights for all examples, the correct F0 of the melody path was found in on average 63% of frames, ranging from 34% in the worst case to a best case of 85%. Compared to the baseline of taking the strongest F0 in every frame, the additional knowledge integration leads to an average increase of 11%. Additionally, even for pieces where the improvement in terms of percentage correct frames by the path finding algorithm is only small, the overall melody contour is significantly smoother. The

removal of short tones leads only to a very small increase in an overall rating of correct frames, but is not unlikely to have a more significant influence perceptually. This increased smoothness of the melody contour is more obvious in a tone based evaluation, where the number of additional tones is reduced by more than half compared to the baseline performance based on F0 strength only.

3.4. Solo Instrument Present/Absent

After estimating the most likely path for the melody, it is still necessary to distinguish between sections where the solo instrument is playing or silent. The path finding algorithm allows for silent states, but in its current state repeatedly follows the accompaniment when the solo instrument is absent for more than a few frames. In the examples used here, the solo instrument is present more often than silent (74% present, 26% silent), and since this is likely to be the case with many musical pieces, it might be favourable to choose a setting that favours the present state for best overall results.

Different smoothing intervals from 0 to 10 sec and different minimal durations of 1 to 30 sec for the presence of the solo instrument were investigated. The different settings had only small influences, unless the smoothing interval was very short and the minimal presence long, in which case the results deteriorated and too many frames were assigned to the silent state. The best parameter setting for overall accuracy uses a smoothing interval of 2 sec (86 frames) and a minimum present time of 10 sec (431 frames). With this setting, 77% of frames were assigned to the correct state (65% if normalised by the relative number of silent compared to present frames). The path finding search alone already assigns the silent state more often during sections where the solo instrument is actually silent. Using the explicit estimation of silent frames improved the accuracy for silence estimation from 24% using only the path based search to 39% correctly assigned silent frames. But as silent frames form only around one quarter of all frames, the improvement for the overall recognition accuracy was only 3% when evaluated on a frame-based basis. Using a tone-based evaluation an additional 16% of superfluous tone were suppressed using the explicit search for silent sections (see Table 2).

	strongest F0	path	path+silence
correct frames	40%	51%	54%
tones found	78%	76%	72%
spurious tones	321%	135%	117%

Table 2: Summary MIDI based evaluation (based on complete examples).

3.5. Realistic Recordings

Two short excerpts from realistic recordings – the beginnings of Mozart's clarinet concert and a violin sonata by Bach – were manually annotated with the correct F0s. Both were pieces which were also included in the main MIDI based evaluation, so that a direct comparison is possible.

Using the same knowledge weight combination that was used for the MIDI generated data, F0s were correctly identified in 56% of frames for the violin example, a result as good for the realistic recording as for the equivalent section of MIDI based audio. Accuracy for the clarinet example was 76% correctly labelled frames, around 5% higher for the realistic recording than the MIDI generated example. All parameters used within the system were optimised on MIDI generated data, and different parameter combinations might be favourable for realistic recordings. Adjusting the parameters for the realistic recordings led to 10%-15% more correctly labeled frames, but it seems premature to draw further conclusions from only two examples.

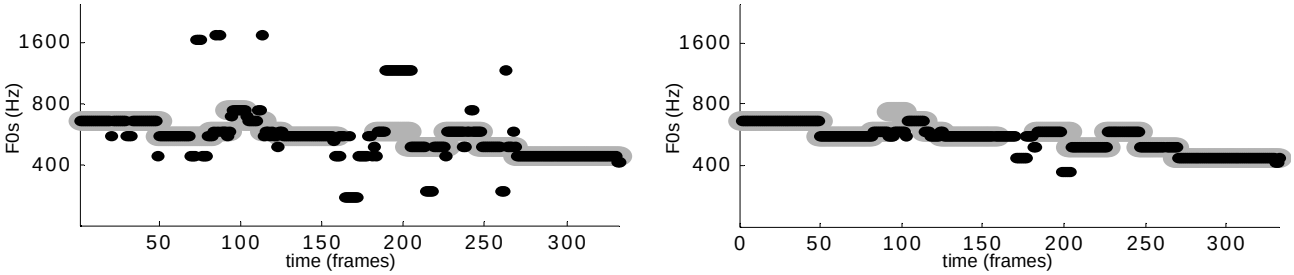
Even though the realistic examples were limited, results seem very promising. Although additional difficulties can normally expected in realistic recordings, such as reverberation and stronger vibrato, accuracies were if anything higher compared to the MIDI generated data (see Figure 2).

4. Conclusion and Future Work

Using the strongest F0 at every frame without further processing can be seen as a baseline performance, and led to an accuracy of 40% correctly estimated F0s on a frame-by-frame basis with no distinction between sections where the solo instrument is actually present or only the accompaniment is playing. The proposed system improved results to an accuracy of 54% and also led to overall smoother melody contours, reducing the number of spurious tones to nearly a third compared to the baseline performance.

A relatively weak point of the system was the estimation of the presence or absence of the solo instrument, which is likely be caused by the low results from the instrument identification process. Since we previously obtained better results for instrument identification using realistic recordings, it can be suggested that some of the difficulties encountered were at least to some extent caused by the MIDI generated test data, and not the system. This finding is supported by the overall better results obtained for two short excerpts from realistic recordings. While MIDI generated audio has the strong advantage of automatically providing a 'true' solution which can be used for evaluation purposes, it does not necessarily provide a good evaluation basis in terms of the acoustics generated.

Additional knowledge sources which could benefit the system might include an instrument recognition model trained on accompaniment. Instead of using only the likelihood that a specific F0 was produced by the solo instrument, this could be weighted against the likelihood that the same F0 was produced by the accompaniment. Higher level knowledge in terms of short melodic phrases,



a) Melody based on strongest F0

b) Melody based on knowledge integrating path finding

Figure 2: Analysis of a realistic recording (the beginning of Mozart's clarinet concerto), showing manually annotated F0s (gray) and estimated melody (black).

which are learned during the analysis of an example, might also help to more robustly identify recurring motifs and phrases. Expanding the instrument basis, especially to include the singing voice, would certainly be interesting. It would also lead to a wider range of music that could be analysed, as pop and rock music is often dominated by a vocalist.

5. References

[1] Arulampalam, M.S., Maskell, S., Gordon, N. & Clapp, T. "A tutorial on particle filters for online nonlinear/non-Gaussion Bayesian tracking", *IEEE Transactions on Signal Processing, Vol. 50, No. 2*, pp. 174-188, 2002

[2] Berenzweig, A.L., & Ellis, D.P.W. "Locating singing voice segments within music signals", *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001

[3] Classical Music Archives, www.classicalarchives.com

[4] Eggink, J. & Brown, G.J. "A missing feature approach to instrument identification in polyphonic music", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 553-556, 2003

[5] Eggink, J. & Brown, G.J. "Instrument recognition in accompanied sonatas and concertos", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, in print, 2004

[6] Goto, M. "A predominant-F0 estimation method for CD recordings: MAP estimation using EM algorithm for adaptive tone models", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3365-3368, 2001

[7] Klapuri, A.P. "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness", *IEEE Transactions on Speech and Audio Processing, Vol. 11, No.6*, pp. 804-816, 2003

[8] Meek, C. & Birmingham, W.P. "The dangers of parsimony in query-by-humming applications", *Proceedings of the International Conference on Music Information Retrieval*, 2003

[9] Raphael, C. "Automatic transcription of piano music", *Proceedings of the International Conference on Music Information Retrieval*, 2002

Acknowledgements

Jana Eggink acknowledges the financial support provided through the European Community's Human Potential Programme under contract HPRN-CT-2002-00276, HOARSE. Guy J. Brown was supported by EPSRC grant GR/R47400/01 and the MOSART IHP network.