Melody Description and Extraction in the Context of Music Content Processing

Emilia Gómez
MTG-IUA, Universitat Pompeu Fabra, Barcelona, Spain
emilia.gomez@iua.upf.es, http://www.iua.upf.es/mtg/

Anssi Klapuri
Signal Processing Laboratory, Tampere University of Technology, Finland
klap@cs.tut.fi, http://www.cs.tut.fi/sgn/arg/

Benoît Meudic
Music Representation Team, IRCAM, Paris, France
benoit.meudic@ircam.fr, http://www.ircam.fr

A huge amount of audio data is accessible to everyone by on-line or off-line information services and it is necessary to develop techniques to automatically describe and deal with this data in a meaningful way. In the particular context of music content processing it is important to take into account the melodic aspects of the sound. The goal of this article is to review the different techniques proposed for melodic description and extraction. Some ideas around the concept of melody are first presented. Then, an overview of the different ways of describing melody is done. As a third step, an analysis of the methods proposed for melody extraction is made, including pitch detection algorithms. Finally, techniques for melodic pattern induction and matching are also studied, and some useful melodic transformations are reviewed.

1 Introduction

Music content processing is a topic of research that has become very important in the last few years. Many researchers have been developing techniques intended to automatically describe and deal with audio data in a musically meaningful way.

In this context, melody plays a major role. "*It is melody that enables us to distinguish one work from another. It is melody that human beings are innately able to reproduce by singing, humming, and whistling. It is melody that makes music memorable: we are likely to recall a tune long after we have forgotten its text.*" (Selfridge-Field, 1998). Melody together with rhythm, harmony, timbre and spatial location make up the main dimensions for sound description. The importance of melody for music perception and understanding reveals that within the concept of melody there are many aspects to consider, since it carries implicit information regarding harmony and rhythm. This fact complicates its automatic representation, extraction and manipulation, which are the subjects of this article.

We review the different techniques intended to deal with these issues. First, in section 2, we define what melody is, how it can be defined and which are the main aspects to be considered. In section 3, we see how melody can be represented and stored, and then we review techniques to automatically extract this representation (section 4). When thinking about melodic analysis, one important issue is to locate patterns and to group notes. This is useful for many tasks related to search and retrieval, navigation, similarity analysis, browsing, or those that perform a structural analysis. This issue is covered in section 5. Finally, in section 6, some musical transformations are presented, which are dependent on the melodic content of the sound.

2 Melody definition

In this section we review different ways of defining and describing a melody. There is no agreed definition of melody and different ones can be found in the literature, each one considering a different aspect of the melodic characterization.

Melody has been defined as an auditory object that emerges from a series of transformations along the six dimensions: pitch, tempo, timbre, loudness, spatial location, and reverberant environment (Kim et al., 2000). This might be the most general definition.

Accepted for publication on May 9th 2002

Sometimes melody is considered as a pitch sequence (what we call *"melody as an entity"* trying to answer the question *"Which is the melody of this audio excerpt?"*). This definition can be found in (Solomon, 1997)*: a combination of a pitch series and a rhythm having a clearly defined shape*, and in Grove Music (Ringer, 2002): p*itched sounds arranged in musical time in accordance with given cultural conventions and constraints.* Goto (2000) (1999) also considers the melody as a sequence of pitches, the most predominant ones in the middle and high registers, in opposition to the bass line, found in the low register.

Melody can also be defined as a set of attributes (what we call *"melody as a set of attributes"* trying to answer the question *"which are the melodic features of this audio excerpt?"*) that characterize the melodic properties of sound. Several attributes to be considered are for example key, melodic profile, melodic density (the degree of melodic activity), and interval distribution or tessitura (pitch range).

Sometimes melody is also associated with the concept of unity: *an arrangement of single tones into a meaningful sequence*. This definition is close to the concept of phrase.

In (Educational Dictionary-Thesaurus, 2000), melody is defined as a *set of musical sounds in a pleasant order and arrangement*, as a theme (*sequence of single tones organized rhythmically into a distinct musical phrase or theme*), and as a song, tune, or phrase (*a short musical composition containing one or a few musical ideas*).

As in the last definition, melody can also be related to the concept of tune or predominant theme, that is usually defined as something recurrent, neither too short nor too long, where all the notes are usually played by the same instrument, and it is often located in the high register. "A t*heme is an excerpt from an arbitrary portion of a work (not necessarily the beginning) which enjoys the greatest melodic importance*" (Selfridge-Field, 1998). Other related concepts are the *Prototypical* melody (if every statement of a theme is slightly varied, find the prototype behind), the *Motive (*rhythmic and/or melodic fragments characteristic of a composer, improviser, or style) and the *Signature* (term for motives common to two or more works of a given composer (Cope, 1998)). These notions show the fact that melody, apart from being a sequential phenomenon, has also a hierarchical structure (motive, phrase, theme, etc.).

For our purposes the most appropriate viewpoints are the ones that consider a melody as a pitch sequence or as a set of attributes. However we mainly concentrate on the pitch sequence viewpoint.

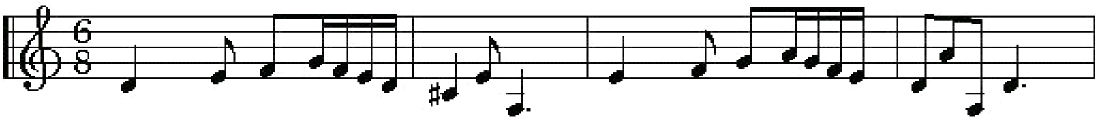
3 Melody representation

In this section, we study melodic description schemes that have been recently proposed. According to Lindsay (1996), a melody representation scheme should have the following properties: compactness (so that the representation can be easily stored), expressiveness (more adapted to a query by humming system, the ability to retain the expressive qualities of the singing voice) and portability (in this type of systems, this mean that the representation must be adaptable to many types of inputs and objects it seeks to match).

Most of these description schemes are considered score-oriented systems, where the melody is mainly described by pitch information. For example, the system proposed by Mc Nab et al. (1996) uses the note pitches as the information for melodic transcription.

As a step forward from pitch information, pitch contour information has also been used in several applications such as query by humming, similarity matching or melodic classification (Lindsay, 1996) (Kim et al., 2000) (Blackburn, 2000). A pitch contour describes a series of relative pitch transitions, an abstraction of a sequence of notes. It has been found to be more significant to listeners in determining melodic similarity, and it also includes rhythm information. This type of description is also found in (Nettheim, 1992) and (Lemström & Laine, 1998), where both pitch and duration sequences are used to define the melody (see Figure 1). Different degrees of precision have been used in representing pitch intervals, varying from up/down discrimination to a semitone precision and even beyond (distinguishing enharmonic differences). The amount of other parameters in processing the melody is heterogeneous, as well.

The earliest approaches disregarded timing information completely, but more recent studies have shown that durational values may sometimes outweight pitch values in facilitating melody recognition (Selfridge-Field, 1998). Duration information is also used by Nettheim (1992) and Lemström & Laine (1998). Smith et al. (1998) studied the length of a melodic query sequence needed to uniquely define a melody in a database of 9400 melodies. Different pitch interval precisions (interval, approximate interval, contour) and with/without durational values were tried. As an obvious result, the more precise the representation is, the shorter query excerpt is needed (about five notes for the most precise). More importantly, however, contour matching together with durational values was more definitive than exact pitch intervals only.



	[293.66, 329.63, 349.23, 392, 349.23, 329.63, 293.66, 277.18, 329.63, 220, 329.63, 698.46, 783.99, 440, 783.99, 349.23, 329.63, 293.63, 440, 220, 293.66]
Duration sequence (multiples of the eighth note duration)	[2, 1, 1, 0.5, 0.5, 0.5, 0.5, 2, 1, 3, 2, 1, 1, 0.5, 0.5, 0.5, 0.5, 1, 1, 1, 3]
	[2, 1, 2, -2, -1, -2, -1, 3, -7, 7, 1, 2, 2, -2, -2, -1, -2, 7, -1, 2, 5]

Figure 1: Melody Description Example.

In all the mentioned representations, melody is considered as an entity (consisting of pitch or contour and duration series). Other information, such as key or scale information can also be taken into account. This is the case in the MPEG-7 melody description scheme that can be found in (MPEG Working Documents, 2001) and (MPEG-7 Schema, 2001), and that is explained in (Lindsay & Herre, 2001).

MPEG-7 proposes a melodic description scheme that includes melody as sequence of pitches or contour values, plus some information about scale, meter, beat and key (see Figure 2).

In MPEG-7 scheme, the *melodic contour* uses a 5-step contour (from –2 to +2) in which intervals are quantized, and also represents basic rhythm information by storing the number of the nearest whole beat of each note, which can dramatically increase the accuracy of matches to a query. This contour has been found to be inadequate for some applications, as melodies of very different nature can be represented by similar contours. One example is the case of having a descendant chromatic melody and a descendant diatonic one. Both of them have the same contour although their melodic features are very dissimilar.

For applications requiring greater descriptive precision or reconstruction of a given melody, the *Melody* Description Scheme (or Melody DS) supports an expanded descriptor set and higher precision of interval encoding. Rather than quantizing to one of five levels, the precise pitch interval (with cent or greater precision) between notes is kept. Precise timing information is stored by encoding the relative duration of notes defined as the logarithm of the ratio between the differential onsets, following the next formula:

$$Noterelduration[n] = \begin{cases} \log_2\left(\dfrac{Onset[n+1]-Onset[n]}{Onset[n]-Onset[n-1]}\right) & \text{for } n \geq 2 \\ \log_2\left(\dfrac{Onset[2]-Onset[1]}{0.5}\right) & \text{for } n = 1 \end{cases}$$

In addition to these core descriptors there are a series of optional descriptors such as lyrics, key, meter, and starting note, to be used as desired for a given application.

This expanded description does not take into account silence parts that sometimes play an essential role for melodic perception. Except for phonemes, the description can be extracted from the score, if available, and no signal level attributes are necessary.

Accepted for publication on May 9th 2002

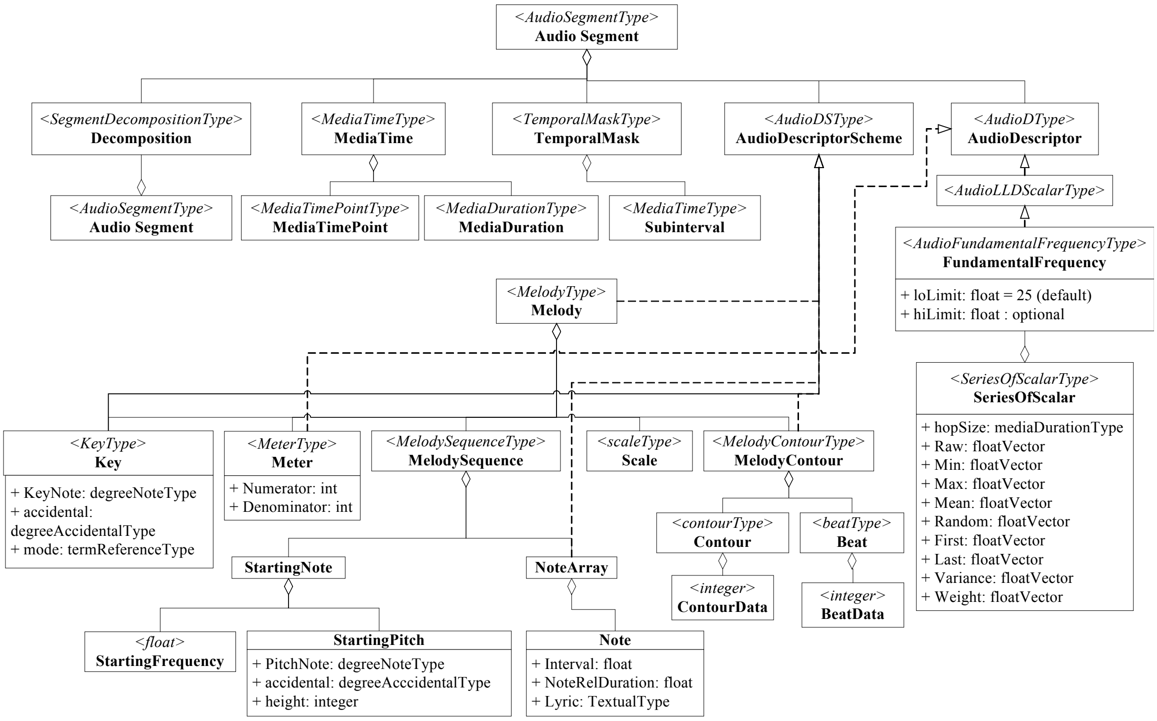


Figure 2: MPEG-7 Melody Description Scheme.

Polyphonic representations allow the encoding of entire musical scores. Perhaps the most common polyphonic representation in computers and between electronic musical instruments is the Musical Instrument Digital Interface (MIDI). According to Lemström and Laine (1998), *MIDI has certain well-known drawbacks: In a conversion process, many important musical properties are not included. Enharmonic notes, bar lines, dynamic markings and almost all information on musical structure are lost when transcribed to MIDI*. Several authors have considered improvements to the limitations of MIDI. One of the most notable among them is the *Kern* representation protocol, proposed by Huron in the context of his music analysis software environment called Humdrum Toolkit (Huron, 1993) (Huron, 1997).

De Cheveigné (2000) has proposed a note-lattice structure for describing melody designed to accommodate both monophonic and polyphonic pitch information, derived from MIDI files or computed from a waveform. The descriptor consists of a lattice of nodes, each one representing a "note" labeled with basic information such as duration, pitch, timbre, estimation reliability, etc. Thus the melody is described by the node transitions. If the score is ambiguous as to what constitutes the melodic line, or if the audio analysis data proposes alternative voices and accompaniments, the lattice supports parallel candidate melodic lines (possibly weighted according to their likelihood). These can be matched in parallel, with possible crossovers from one line to the other.

All these description schemes have been successfully used for different applications and in different contexts. However, if we want to share them they should be integrated, which is not easy. One possibility would be to define a data container general enough to be used in the different applications. The features to be included should include:

- Melodic attributes derived from a numerical analysis of pitch information:
    - Number of notes
    - Tessitura
    - Interval distribution
    - Melodic profile
    - Melodic density
- Melodic attributes derived from a musical analysis of the pitch data:
    - Key information
    - Scale information (scale type: diatonic, chromatic, pentatonic, etc)
    - Cadence information
- Melodic attributes derived from a structural analysis:
    - Motive analysis

- Repetitions
- Patterns location
- Phrase segmentation (also including musical knowledge).

These fields (see explanation at 4.3) can be considered as mid and low-level features. Some meaningful and interesting higher-level features (as for example emotional descriptors) may be derived from them. For example: an ascending profile and a medium-large density could be mapped to a certain subjective category.

4 Melody description extraction

In this section, we review different techniques for automatically extracting the fields of the description scheme. We also present operations, such as fundamental frequency detection and pattern analysis methods, which do not directly result into a melodic feature but that are needed as intermediate steps.

Due to the importance of pitch, or fundamental frequency, detection in speech and music processing, there are a lot of relevant references. Several surveys and evaluation studies are available, for example (Hess, 1983) (Road, 1996) (Romero & Cerdá, 1997) and (Klapuri, 2000).

4.1 Fundamental frequency estimation for monophonic sounds

The first algorithms used for fundamental frequency[1] detection of musical signals were taken from the speech literature (Hess, 1983). Recently new methods have been specifically designed for music.

The algorithms available can be classified in different ways. For example, it is useful the distinguish them according to their processing domain and separate the *time-domain* from the *frequency-domain* algorithms. This separation is not always that clear, since some of the algorithms can be expressed in both (time and frequency) domains. This is the case of the Autocorrelation Function (ACF) method. A classification of the frequency domain methods is based on distinguishing between *spectral place* and *spectral interval* algorithms (Klapuri, 2000). The *spectral place* algorithms, as the ACF method and the cepstrum analysis, weight spectral components according to their spectral location. Other systems, such as those based on envelope periodicity or spectrum autocorrelation computation, are based on measuring the *spectral intervals* between components. Then, the spectrum can be arbitrary shifted without affecting the output value. These algorithms work relatively well for sounds that exhibit inharmonicity, because intervals between harmonics remain more stable than the places for the partials.

All the fundamental frequency estimation algorithms give us a measure corresponding to a portion of the signal (analysis frame). According to Hess (1983), the fundamental frequency detection process can be subdivided into three main steps that are passed through successively: the *preprocessor*, the *basic extractor*, and the *postprocessor* (se Figure 3). The basic extractor performs the main task of measurement: it converts the input signal into a series of fundamental frequency estimates. The main task of the pre-processor is data reduction in order to facilitate the fundamental frequency extraction. Finally, the postprocessor is a block that performs more diverse tasks, such as error detection and correction, or smoothing of an obtained contour.

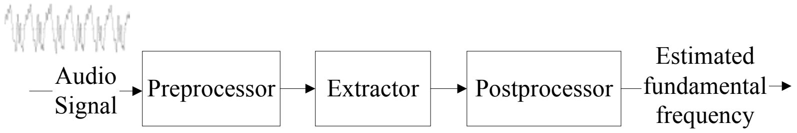


Figure 3: steps of the fundamental frequency detection process.

Time-domain algorithms

These algorithms try to find the periodicity of the input sound signal in the time domain.

- *Zero-crossing rate (ZCR):* ZCR is among the first and simplest techniques for estimating the frequency content of a signal in time domain, and consists in counting the number of times the signal crosses the 0-level reference in order to estimate the signal period. This method is very simple and inexpensive but not very accurate when dealing with noisy signals or harmonic signals where the partials are stronger than the fundamental. The value of ZCR has also been found to correlate strongly with the spectral centroid, which is the first moment of the spectral power distribution, and

[1] We consider the term “fundamental frequency” as a signal feature in opossition to “pitch”, which is a perceptual measure.

in fact it relates more with timbre than with pitch. Thus, this method and its variants are not very much used for fundamental frequency estimation.

– *Autocorrelation (ACF):* Time-domain Autocorrelation function based algorithms have been among the most frequently used fundamental frequency estimators. The ACF of a sequence *x(n)* of length *K* is defined as:

$$r(n) = \frac{1}{K} \sum_{k=0}^{K-n-1} x(k) \cdot x(k+n)$$

The maximum of this function corresponds to the fundamental frequency for periodic signals. One good example of fundamental frequency detection using correlation methods is (Medan et al., 1991), where it maximizes the cross-correlation function over the range of feasible pitch values. This algorithm was tested on synthetic and real speech data covering a large range of speakers and a full range of pitch frequencies. A latter example of algorithms based on cross-correlation is the Robust Algorithm for Pitch Tracking by Talkin (1995). The algorithm uses a two-step calculation of the normalized cross correlation function (NCCF) between successive segments of the input signal. By using cross correlation instead of autocorrelation functions, these algorithms achieve a relatively good time resolution even for low-pitched sounds.

The autocorrelation can also be computed in the frequency domain (Klapuri, 2000). First, *x(k)* is zero-padded to twice its length and transformed into the frequency domain using the short-time Fourier transform (STFT). Then, the square of the magnitude spectrum is obtained, $|X(k)|^2$, and transformed back to the time domain. The autocorrelation function can be expressed as:

$$r(n) = \frac{1}{K} \sum_{k=0}^{K-1} \left[|X(k)|^2 \cdot \cos\left(\frac{2\pi nk}{K} \right) \right]$$

According to the classification made by Klapuri (2000) since the ACF weights spectral components according to their spectral location, these systems can be called *spectral place* type fundamental frequency estimators. In these algorithms, “twice-too low” octave errors are likely to occur, since integer multiples of the fundamental frequency n_0 also have positive weights at the harmonics frequencies. “Too high” octave errors are not probable, since in that case off harmonics get a negative weight.

ACF based fundamental frequency detectors have been reported to be relatively noise immune (Romero & Cerdá, 1997) but sensitive to formants and spectral peculiarities of the analyzed sound (Klapuri, 2000).

– *Envelope periodicity:* The idea behind this model is derived from the observation that signals with more than one frequency component exhibit periodic fluctuations (beatings) in its time domain amplitude envelope. The rate of these fluctuations depends on the frequency difference of each two frequency components. In the case of a harmonic sound, interval f_0 will dominate and the fundamental frequency is clearly visible in the amplitude envelope of the signal. This algorithm is *spectral interval* oriented and implicitly applies spectral smoothing, because the amplitude envelope computation filters out higher amplitude harmonic partials. It follows that smooth spectrum causes strong beating. A third property of EP models is that they are phase sensitive in summing up the harmonic partials (Klapuri, 2000)..

The most recent models of human pitch perception calculate envelope periodicity separately for distinct frequency bands and then combine the results across channels (Meddis & Hewitt, 1991). These methods attempt to estimate the perceived pitch, not pure physical periodicity, in acoustic signals of various kinds. The algorithm proposed by E. Terhardt represents an early and valuable model (Terhardt, 1979) and (Terhardt et al., 1981). Except for the simplest algorithms, that only look for signal periodicity, “perceived pitch” estimators use some knowledge about the auditory system when preprocessing, extracting, or post processing data. Thus, they could be considered as pitch estimators. However, since the psychoacoustic knowledge is only applied to improve the periodicity estimation and no complete model of pitch perception is applied, they do not explain some auditory phenomena.

– *Parallel processing approach:* the fundamental frequency detector defined by Gold and later modified by Rabiner was designed to deal with speech signals (Gold & Rabiner, 1969) (Rabiner & Schafer, 1978). This algorithm has been successfully used in a wide variety of applications and it is based on purely time domain processing. The algorithm has three main steps (shown in Figure 4):

1. The speech signal is processed so as to create a number of impulse trains that retain the periodicity of the original signal and discard features that are irrelevant to the pitch detection method. This can be considered as the pre-processing part of the algorithm.
2. Simple estimators are used to detect the period of these impulse trains.

3. All the estimates are logically combined to infer the period of the speech waveform.

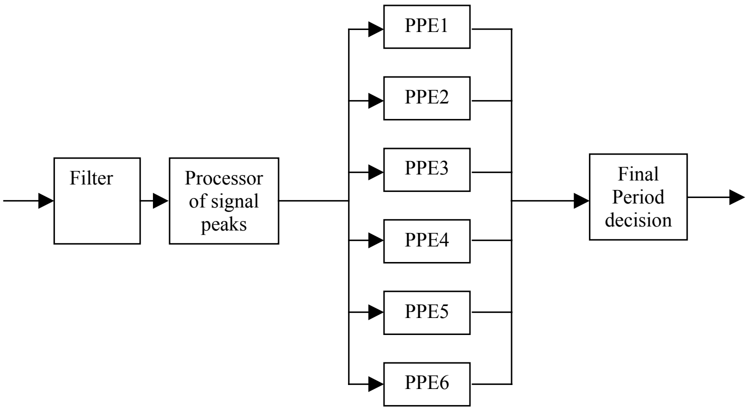


Figure 4: parallel processing approach.

The approach of having several processes working in parallel is unique and interesting here. This path has not been much explored, but it is plausible from the human perception point of view, and it might be very fruitful. Bregman remarks: *"I believe that there is a great deal of redundancy in the mechanisms responsible for many phenomena in perception"* (Bregman, 1998). Several different processes analyze the same problem, and when one fails, the other succeeds. This algorithm has a very low computational complexity and it gives relatively good performances.

Frequency-domain algorithms

These algorithms search for the fundamental frequency using the spectral information of the signal, obtained by an STFT or another transform.

- *Cepstrum analysis:* Noll (1967) introduced this idea for pitch determination of speech signals. The cepstrum pitch detection algorithm was the first short-term analysis algorithm that proved realizable on a computer. This algorithm served as a calibration standard for other, simple, algorithms. The cepstrum is the inverse Fourier transform of the logarithm of the power spectrum of the signal. By the logarithm operation, the source and the transfer functions are separated. Consequently, the pulse sequence originating from the periodicity source reappears in the cepstrum as a strong peak at the "quefrency" (lag) T_0, which is readily discovered by the peak-picking logic of the basic extractor. Some modifications were made to the algorithm in order to speed up the analysis. Nevertheless, these modifications elucidate the role of certain pre-processing methods, such as center clipping.

  Cepstrum fundamental frequency detection has close model level similarity with autocorrelation systems. The only difference to frequency domain ACF calculations is that the logarithm of the magnitude spectrum is used instead of second power. Cepstrum can also be considered as a *spectral place* type algorithm, with "too low" octave errors. Unlike ACF systems, cepstrum pitch estimators have been found to perform poorly in noise and to have good performances with formants and spectral peculiarities of the analyzed sounds (Klapuri, 2000) (Romero & Cerdá, 1997).
- *Spectrum autocorrelation*: The idea of these methods is derived from the observation that a periodic but non-sinusoidal signal has a periodic magnitude spectrum, the period of which is the fundamental frequency. The goal is to detect the period of the magnitude spectrum using its autocorrelation function. It can be called *spectral interval type* fundamental frequency estimator (Klapuri, 2000). "Too low" octave errors are not probable since there is no spectral periodicity at half the fundamental frequency rate. But "twice too high" octave errors are likely to occur, since doubling the true spectral period picks every second harmonic of the sound. A nice implementation of this principle can be found in (Lahat et al., 1987).
- *Harmonic Matching Methods:* These algorithms identify the periodicity from the spectral peaks of the magnitude spectrum of the signal. Once these peaks are identified, they are compared to the predicted harmonics for each of the possible candidate note frequencies, and a fitting measure is developed. One of the first developed methods was (Piszczalski & Galler, 1979), which tries to find the best fitting harmonic number for each component pair of the spectrum. A particular fitness measure, named "Two Way Mismatch" procedure, is described in (Maher & Beauchamp, 1993). For each fundamental frequency candidate, mismatches between the harmonics generated and the measured partials frequencies are averaged over a fixed subset of the available partials. A weighting scheme is

used to make the procedure robust to the presence of noise or absence of certain partials in the spectral data. The discrepancy between the measured and predicted sequences of harmonic partials is referred as the *mismatch error*. The harmonics and partials would “live up” for fundamental frequencies that are one or more octaves above and below the actual fundamental; thus even in the ideal case, some ambiguity occurs. In real situations, where noise and measurement uncertainty are present, the mismatch error will never be exactly zero.

The solution presented in (Maher & Beauchamp, 1993) employs two mismatch error calculations. The first one is based on the frequency difference between each partial in the measured sequence and its nearest neighbor in the predicted sequence (see Figure 5). The second is based on the mismatch between each harmonic in the predicted sequence and its nearest partial neighbor in the measured sequence. This two-way mismatch helps to avoid octave errors by applying a penalty for partials that are present in the measured data but are not predicted, and also for partials whose presence is predicted but which do not actually appear in the measured sequence. The TWM mismatch procedure has also the benefit that the effect of any spurious components or partial missing from the measurement can be counteracted by the presence of uncorrupted partials in the same frame.

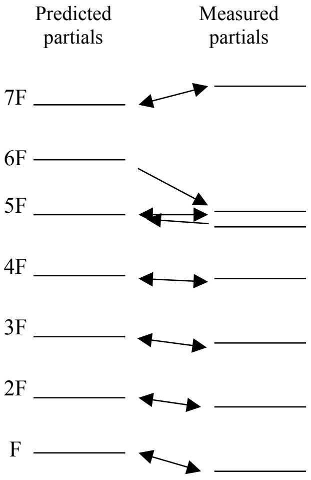


Figure 5: TWM procedure

The two error measurements are computed as follows:

- Predicted-to-measured mismatch error:

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

where a_n, f_n correspond to the amplitude and frequency of the predicted partial number *n*, A_{max} is the maximum amplitude, and Δf_n is the difference between the frequency of the predicted partial and its closest measured partial.

- Measured-to-predicted mismatch error:

$$Err_{m\to p} = \sum_{k=1}^{K} E_{\omega}(\Delta f_k, f_k, a_k, A_{\max}) = \sum_{k=1}^{K} \Delta f_k \cdot (f_k)^{-p} + (\frac{a_k}{A_{\max}}) \times \left[q\Delta f_k \cdot (f_k)^{-p} - r\right]$$

where a_k, f_k correspond to the amplitude and frequency of the measured partial number k, A_{max} is the maximum amplitude, and Δf_k is the difference between the frequency of the measured partial and its closest predicted partial.

The total error for the predicted fundamental frequency is then given by:

$$Err_{total} = Err_{p\to m} / N + \rho \cdot Err_{m\to p} / K$$

The parameters (*p, q, r, ρ,* etc*)* are set empirically. This is the method used in the context of the Spectral Modeling Synthesis system (from now on SMS) (Cano, 1998) including some improvements, as having pitch dependent analysis window, a selection of spectral peaks to be used, and an optimization in the search for fundamental frequency candidates.

Another harmonic matching method is the one described in (Doval & Rodet, 1991), based on a maximum likelihood for the fundamental frequency. After detecting the maxima of the spectrum, the probability that a maximum corresponds to a partial rather than to a noise is computed for each detected peak. To find the optimal solution, it proceeds in two steps: first, it computes the interval that contains the optimal solution, and then it obtains the precise optimal value for the fundamental frequency within this interval. To determine which interval contains the optimal solution, the value of likelihood is computed for each of the selected intervals of the frequency axis. Then, more precise optimal values are obtained using a regression on the frequencies of the signal partials that are matched with the harmonic partials.

- *Wavelet based algorithms*: the wavelet transform (WT) is a multiresolution, multi-scale analysis that has been shown to be very well suited for music processing because of its similarity to how the human ear processes sound. In contrast to the STFT, which uses a single analysis window, WT uses short windows at high frequencies and long windows for low frequencies. This is the spirit of the constant Q ($\frac{\Delta f}{f_c}$) frequency analysis. Some wavelet based fundamental frequency algorithms have been proposed for speech analysis. They can be adapted to music analysis, as the one proposed by Jehan (1997) for voice signals. The idea is to filter the signal using a wavelet with derivative properties. The output of this filter will have maxima where GCI (*Glottal Closure Instant*) or zero crossings happens in the input signal. After detection of these maxima, the fundamental frequency will be estimated as the distance between consecutive maxima. This filtering function will combine the bandwidth properties of the wavelet transform at different scales.
- *Bandwise processing algorithm*: following with the idea of the constant Q frequency analysis, Klapuri (2000) has proposed an algorithm for periodicity analysis that calculates independent fundamental frequencies estimates at separate frequency bands. Then, these values are combined to yield a global estimate. This solves several problems, one of which is inharmonicity. In inharmonic sounds, as stretched strings, the higher harmonics may deviate from their expected spectral positions, and even the intervals between them are not constant. However, we can assume the spectral intervals to be piece-wise constant at narrow enough bands. Thus we utilize spectral intervals to calculate pitch likelihoods at separate frequency bands, and then combine the results in a manner that takes the inharmonicity into account. Another advantage of bandwise processing is that it provides robustness in the case of badly corrupted signals, where only a fragment of the whole frequency range is good enough to be used. A single fast Fourier transform is needed, after which local regions of the spectrum are separately processed. The equalized spectrum is processed in 18 logarithmically distributed bands that extend from 50Hz to 6000Hz. Each band comprises a 2/3-octave wide region of the spectrum that is subject to weighting with a triangular frequency response. Overlap between adjacent bands is 50%, which makes the overall response sum to unity. Fundamental frequency likelihoods are calculated at each band, and the values are combined taking into account that fundamental frequency can increase as a function of the band center frequency for string instruments. Some improvements were made to provide robustness in interference, where pitch is observable only at a limited band, and to adapt to signals containing a mixture of several harmonic sounds.

In Table 1 we summarize the listed algorithms. The information about the performances has been extracted from different sources such as (Klapuri, 2000) (Romero & Cerdá, 1997) or from the authors' related work.

Pitched/unpitched decision

Fundamental frequency detection algorithms assume that a fundamental frequency is present, and it is not always the case. We could have segments where no pitch is found, as for instance in silences, percussion or noise segments. A segmentation process should distinguish between voiced and unvoiced periods, so that the fundamental frequency detection will be only performed for pitched parts of the signal.

However, most of the techniques used for pitched/unpitched segmentation already use the estimated fundamental frequency to decide whether this information is valid or corresponds to an unpitched signal, in addition to other features computed from the signal.

One example appears in the SMS context (Cano, 1998), where the portions of a sound that cannot be well represented with the harmonic model are considered as a noisy component. There is a strict time segmentation with routines that use the error measure of the TWM procedure along with other measures that are easily computed as part of the SMS analysis: zero crossing, energy, noisiness and harmonic distortion.

Preprocessing methods

The main task of a preprocessor is to suppress noise and to enhance the features that are useful for fundamental frequency estimation. The fundamental frequency signal from a physical vibrator is usually first filtered by passing through resonance structures (e.g. body of an instrument) and by the environment, and then linearly superimposed with other co-occurring sounds and noise. The first type of interference can be called *convolutive noise*, and the latter type *additive noise*. Both should be removed to reveal the underlying fundamental frequency.

Convolutive noise suppression is usually called spectral whitening, since it aims at normalizing away all spectral peculiarities of the sound source and the environment, however, leaving the spectral fine structure (fundamental frequency) intact. Inverse linear predictive filtering is a common way of performing this task. Suppression of additive noise is usually done by subtracting an estimate of the noise spectrum in power spectral domain. Additive noise spectrum can be estimated e.g. by calculating the median of filterbank output levels over time.

Some of the preprocessing methods used in speech processing are detailed in (Hess, 1983) such as isolation of the first partial, moderate low-pass filtering to remove the influence of higher formants, inverse filtering to obtain the excitation signal, comb filtering to enhance harmonic structures, application of a filterbank, non linear processing in the spectral domain, center clipping (that destroys the formant structure without destroying the periodic structures of the signal), signal distortion by an even nonlinear function, envelope detection, instantaneous-envelop detection, and algorithmic temporal structure investigation. These algorithms can also be used for the fundamental frequency detection of music signals.

Some preprocessing methods have been specifically defined for musical signals. The method proposed by Klapuri applies the principles of RASTA spectral processing (Klapuri et al., 2001) (Hermansky et al., 1993) in removing both additive and convolutive noise simultaneously. After obtaining the signal spectrum, a non-linear transformation is applied to the spectrum X(k) to yield X'(k)=ln{1+J*X(k)}. With a proper scaling of the spectrum, additive noise goes through a linear-like transform, whereas spectral peaks, affected by convolutive noise, go though a logarithmic-like transform. By subtracting a moving average over a Bark-scale spectrum from X'(k), both convolutive and additive noise are suppressed in a single operation.

Method	**Domain**	**Spectral Place**	**Spectral Interval**	**Simplicity**	**Noise Robustness**	**Inharmonicity Robustness**	**Spectral peculiarities Robustness**
ZCR	Time	✓		Very simple			
ACF	Time /Frequency	✓		Simple	Relatively noise immune		Sensitive
EP	Time		✓	Simple			
Rabiner	Time	✓		Relatively simple			
Cepstrum	Frequency	✓		Simple	Poor performance		Relatively immune
Spectrum AC	Frequency		✓	Simple			
Harmonic Matching Methods	Frequency	✓	✓	Quite Complex	Relatively noise immune		Relatively immune
Wavelet based method	Frequency (WT)			Quite Complex	Noise immune		
Bandwise EP	Time	✓	✓	Quite Complex	Rather noise immune	Relatively immune	Relatively immune
Bandwise/ Klapuri	Frequency	✓	✓	Quite Complex	Relatively noise immune	Relatively immune	Relatively immune

Table 1: Fundamental Frequency Algorithm comparison.

Postprocessing methods

The fundamental frequency contour that is the output of the different algorithms is normally noisy and sometimes badly affected by isolated errors, so different methods for correcting them have been proposed.

The most common way to smooth a function is by convolving the input signal with the impulse response of a lowpass filter. Since the smoothing function (window) usually is of a very short length, this convolution can be reduced to the weighted addition of few samples. Since the convolution is linear, we speak of *linear smoothing*. As presented in (Hess, 1983), the low pass filtering removes much of the local jitter and noise, but it does not remove local gross measurements errors, and, in addition, it smears the intended discontinuities at the voiced-unvoiced transitions. Hence, some kind of *non-linear smoothing* might be more appropriate. In (Rabiner et al. 1975) median smoothing is proposed as a non-linear method. They recommend the use of a combination of median and linear smoothing in that order, because median removes short errors, and the linear smoothing removes jitter and noise.

Another approach is described in (Laroche, 1995). The procedure consists in storing several possible values for the fundamental frequency for each analysis frame, assigning them a score (for example, the value of the normalized autocorrelation or the TWM error) that represents the estimation goodness. The goal is to find a "track" that, following one of the estimations for each frame, will have the best score. The total score is computed using the score of the estimations considering penalizations if, for example, an abrupt fundamental change is produced. This optimal "track" can be obtained using dynamic programming techniques or using some Hidden Markov Models (as in the method proposed by Doval and Rodet (1993)). This approach minimizes the abrupt fundamental frequency changes (octave errors, for example) and gives good results in general. Its main disadvantage is that there is a delay in the estimation, as we use past and future fundamental frequency values.

Other postprocessing methods are dependent on the algorithm used for sound analysis and fundamental frequency detection. One example is the approach used in the context of SMS. Fundamental frequency is estimated using spectral peak information. Spectral peaks are computed frame by frame using windowing and FFT analysis. In the windowing procedure, window size is updated depending on the previous fundamental frequency detected. If an estimation error is produced, then the window size for the following frames will not be correctly chosen. To solve this a reanalysis over past frames is performed when needed. Past and future information is also used to choose among candidates with the same TWM error (Cano, 1998) thus smoothing the fundamental frequency function.

Fundamental frequency tracking using other knowledge sources

In this section, we are interested in the use of available meta-data for guiding the fundamental frequency detection and tracking process. In this point, we study the applicability of using content information for refining the process of monophonic fundamental tracking.

Content information has been used in the form of internal sound source models (Kashino et al., 1995). Martin (1996) also used musical rules to transcribe four-voice polyphonic piano pieces. When some assumption about the type of signal is made or when the algorithm is adapted to some of the signal properties, we are also taking advantage of some information that can be considered as context information.

- *Timbre:* We have seen that the different algorithms work better for different sound sources in different conditions. If the algorithm chosen depends on the instrument, we could consider that some timbre (meta-data) information may be used. Also some preprocessing methods can be dependent on the instrument played. Goto discriminates melody and bass lines using different band-pass filters (Goto, 2000) (Goto, 1999). Another approach is to adapt some of the parameters of an algorithm (for example, the TWM algorithm presented in (Maher & Beauchamp, 1993)) to the played instrument (e.g. considering the inharmonicity of the piano or the particularity of the spectra of some instruments). This idea is the basis of some of the multitimbre pitch detectors (see (Goto, 2000) (Goto, 1999) and (Anderson, 1997)).
- *Rhythm:* Rhythm information has not been considered in any fundamental frequency detector. One idea could be to use information about rhythm as note duration for detecting pitch changes, provided rhythmic information could be computed first and independently. On the other hand, fundamental frequency estimation is a useful feature to detect note boundaries.
- *Melody:* Melody description is made mostly using pitch information, so melodic information as a pitch sequence should not be available for fundamental frequency detection tracking. However, if some higher-level melodic information is present (as for example melodic profile, scale, key, etc), this could be used for fundamental tracking. For instance, key or scale information could be used to give a higher expectancy of fundamental frequency values that match the scale.
- *Genre:* Some knowledge about the genre of music that is being played or some musical knowledge could also be used. For example, "atonal" changes should not be found in most of genres, certain styles use only a restricted set of scales, etc.

4.2 Multipitch estimation methods

It is generally admitted that single-pitch estimation methods are not well suited for multipitch estimation (Klapuri et al., 2000) even though some of the algorithms used in monophonic pitch detection can be adapted to simple polyphonic situations. In (Anderson, 1997) it is described how some of the methods applied to monophonic fundamental frequency estimation can be adapted to polyphony. Also, the TWM procedure can be extended to duet separation, as explained in (Maher & Beauchamp, 1993), trying to find two fundamental frequencies that best explain the measured spectral peaks.

Multipitch estimation is oriented towards auditory scene analysis and sound separation: if an algorithm can find the pitch of a sound and not get confused by other co-occurring sounds, the pitch information can be used to separate the partials of the sound from the mixture. Indeed, the most successful multipitch estimation methods have applied the principles known from human auditory organization.

Kashino et al. implemented these principles in a Bayesian probability network, where bottom-up signal analysis could be integrated with temporal and musical predictions (Kashino et al., 1995). Another recent example is that of Wamsley et al. (1999), who use the bayesian probabilistic framework in estimating the harmonic model parameters jointly for a certain number of frames. Godsmark and Brown (1999) have developed a model that is able to resolve melodic lines from polyphonic music through the integration of diverse knowledge. The system proposed by Goto (1999) (2000) is more application-oriented, and it is able to detect melody and bass lines in real-world polyphonic recordings by making the assumption that these two are placed in different frequency regions.

Other methods are listed in (Klapuri et al., 2000), and a system is described following an iterative method with a separation approach. This algorithm operates reasonably accurately for polyphonies at a wide fundamental frequency range and for a variety of sound sources.

The state-of-the-art of multipitch estimators operate reasonably accurately for clean signals, the frame-level error rates progressively increasing from a couple of percent in two-voice signals up to about twenty percent error rates in five-voice polyphonies. However, the performance decreases significantly in the presence of noise, and the number of concurrent voices is often underestimated. Also, reliable multipitch estimation requires significantly longer time frames (around 100 ms) than single-pitch estimation (Klapuri et al., 2000).

4.3 Extracting melody

Several features play a role in the description of melody. Among those features, pitch appears to be the most often used. However, we believe that a melody can also be characterized without this information: in this case, melody will not only be characterized by a pitch sequence (melody as an entity) but also by several properties and aspects relative to this sequence (melody as a set of attributes).

Considering melody as a pitch sequence, several cases make the automatic extraction of melody very difficult (Nettheim, 1992):

- A single line played by a single instrument or voice may be formed by movement between two or more melodic or accompaniment strands.
- Two or more contrapuntal lines may have equal claim as “the melody”.
- The melodic line may move from one voice to another, possibly with overlap.
- There may be passages of figuration not properly considered as melody.

In the following section, we will first consider melody as an entity. We will present some issues and approaches that are related to the extraction of melodic lines from pitch sequences. Then, we will consider some features that could characterize the melodic features of an audio excerpt without providing an explicit pitch sequence.

Melody as a pitch sequence

We have presented above several methods that aimed at tracking pitches. As output, the algorithms provided pitch sequences. In this part, we will present some approaches, which attempt to identify the notes of the pitch sequences that are likely to correspond to the melody. This task can be considered not only for polyphonic sounds, but also for monophonic sounds. Indeed, monophony could have notes that do not belong to the melody (as for example grace notes, passing notes or the case that we had several interleaved voices in a monophony).

In a first step, in order to simplify the issue, one should try to detect note groupings. This would provide heuristics that could be taken as hypothesis in the melody extraction task. For instance, experiments have been done on the way the listener achieves melodic groupings in order to separate the different voices (see (Mc Adams, 1994) and (Scheirer, 2000) p.131).

Other approaches can also simplify the melody extraction task by making assumptions and restrictions on the type of music that is analyzed. Indeed, melody extraction depends not only on the melody definition, but also on the type of music from which we want to extract the melody. For instance, methods can be different according to the complexity of the music (monophonic or polyphonic music), the genre (classical with melodic ornamentations, jazz with singing voice, etc) or the representation of the music (audio, midi etc).

Uitdenbogerd and Zobel (1998) have worked on MIDI files containing channels information. She has evaluated some algorithms that extract from a MIDI file a sequence of notes that could be called the melody. The first algorithm considers the top notes of the sequence as the melody. The others combine both entropy information and structure of the MIDI file. However, the first algorithm gives the best results.

According to (Uitdenbogerd & Zobel, 1998), "*there does not appear to have been much research on the problem of extracting a melody from a piece of music*". Indeed, it appears that very few researches focus on this area in comparison to the interest that is given to other tasks such as melody matching and pattern induction.

Melody as a set of attributes

Several other features than pitch can be used in order to characterize a melody. One can group those features in categories:

- *Harmonic features*: Several features related to harmony can provide information on the melody as the key, the scale, and the position of the cadences. However, the automatic extraction of these features is a very difficult task that requires musical knowledge and presents some ambiguities. For example, in order to detect the scale, we first need to identify which notes belong to the scale and which ones are out of it. Then, there can be more than one possible harmony (key, scale, etc) for the same set of pitches. Some work on harmony analysis and references can be found in (Barthélemy & Bonardi, 2001).
- *Rhythmic features*: Rhythm is an important descriptor for melody. The rhythm extraction issue has been addressed in several articles (references can be found in (Desain & Windsor, 2000) (Scheirer, 2000) or (Gouyon et al., 2002)).
- *Features based on pitch information*: there are some other features that can be computed using pitch information at different levels of analysis (introduced in section 3):
  - Derived from a numerical analysis of pitch information:
    - Number of notes
    - Tessitura: or pitch range ($Tessitura = \max(pitch) - \min(pitch)$)
    - Interval distribution: this is another aspect to consider when describing melodic properties of an audio excerpt, which types of intervals are used.
    - Melodic profile: depending on the pitch contour we could define several types of pitch profiles: ascending, descending, constant, etc.
    - Melodic density: is the degree of melodic activity, which can be defined in relation with the rhythmic distribution of notes.
  - Derived from a musical analysis of the pitch data:
    - Key information
    - Scale information (scale type: diatonic, chromatic, pentatonic, etc)
    - Cadence information: which types of cadences are used.
  - Derived from a structural analysis:
    - Motive analysis
    - Repetitions
    - Patterns location (explained in section 5)
    - Phrase segmentation (also including musical knowledge).

- *Perceptual or subjective features*: These features will be treated as secondary in this article, but it should be interesting to consider them in future work. Some emotional/textual melodic descriptors could be defined, as for example sad, happy, or charming. One interesting approach has been developed by the Department of Speech, Music and Hearing of the Royal Institute of Technology at Stockholm[2]. Some context dependent rules characterizing a music performance have been defined. These rules affect duration, pitch, vibrato and intensity of notes, and can create crescendos, diminuendos, change the tempo or insert pauses between tones. They can be classified into three groups: differentiation rules, grouping rules and ensemble rules. Combinations of these rules define different emotional qualities (fear, anger, happiness, sadness, tenderness, and solemnity).

[2] http://www.speech.kth.se/music/performance

5 Melody pattern induction and matching

This section addresses a very general issue that is encountered in any application manipulating data sequences (music, text, video, data coding, etc). It consists of comparing and measuring the similarity between two different sequences of events. In the following sections, we will associate the concept of melody matching to the notion of pattern induction (or extraction). Indeed, comparing two melodies and extracting a musical pattern from a sequence have common aspects: in both tasks, one needs to perform similarity measurements.

We will present some approaches that could be particularly interesting in the context of melody analysis.

5.1 Non music-oriented approaches

Several non music-oriented disciplines deal with pattern induction (extraction) or pattern matching. However, most of the algorithms that are employed cannot be directly used in a musical context. Indeed, musical patterns must be determined considering various musical dimensions, for example temporal, cognitive, and contextual ones.

For instance, we can consider a data compression algorithm, the Lempel-Ziv algorithm (Ziv & Lempel, 1977), which has already proved to have musical applications (Lartillot et al., 2001). The Lempel-Ziv algorithm proceeds as follows: a sequence of elements is read from the beginning to the end element by element. At each step, the current element, if different from the previous ones, is placed in a database. If the element is already in the database, then the pattern constituted by that element and its following one is considered, etc. The method seems interesting as it filters some patterns among all the possible patterns. The question is: are the detected patterns linked with a musical structure?

One could answer that the motivic and melodic structures are not preserved. Indeed, once a pattern is detected, it cannot be changed by the new information given by the following elements of the sequence, so the positions of the patterns in the sequence are arbitrarily determined by the order in which the sequence is analyzed. Moreover, possible hierarchical relations between the patterns are not taken into account, which seems contradictory with the musical organization of motives. However, one advantage is that the method is incremental and analyses each event of the sequence by considering the only already analyzed ones, which may be linked with the conditions in which we listen to music.

This illustrates the fact that the specificities of music (temporal aspect, polyphonic aspect, hierarchical aspect etc…) must be considered before adapting any algorithm to a musical purpose. Several studies have been done on this subject. For instance, Cambouropoulos (2000) discusses some of the issues relative to the application of string processing techniques on musical sequences, paying special attention to pattern extraction.

5.2 Music-oriented approaches

According to Rowe (2001), p. 168, "*Music is composed, to an important degree, of patterns that are repeated and transformed. Patterns occur in all of music's constituent elements, including melody, rhythm, harmony and texture*". Pattern induction means learning to recognize sequential structures from repeated exposures, followed by matching new input against the learned sequences.

Melodic pattern matching has several uses in interactive music systems and in music content analysis. For instance, Cope (1998) has used patterns to define the stylistic characteristics of a particular composer, and to reproduce algorithmic compositions that resemble the works of a certain composer. Cope employs the term *signature,* used to refer to melodic and rhythmic patterns found repeatedly in a composer's body of work. Rolland and Ganascia have also used pattern induction to analyze the jazz improvisations of Charlie Parker (Rolland & Ganascia, 1996). In their system, patterns in a musical corpus are clustered according to similarity, and each cluster is associated with one prototypical pattern.

In automatic music transcription or musical interaction, the predictive power of recognized musical patterns as originators of expectations would be highly valuable.

The basic problem in musical pattern induction is that the patterns most often appear in varying and transformed forms. In music, it is natural that a melody may be transposed to a different pitch register, or played with slightly different tempo. Also, members of the melodic sequence may be deleted, inserted, substituted, or rhythmically displaced. Several elements may be replaced with single one (consolidation), or vice versa (fragmentation).

Matching varied patterns calls for a proper definition of *similarity.* The trivial assumption that two patterns are similar if they have identical pitches is usually not appropriate. Relaxation of this assumption can be approached by allowing some different notes to appear in the sequences, or by requiring only that the profiles of the two melodies are similar. Most often it is important to consider the note durations along with the pitch values. Sometimes, other features such as dynamics, timbre or note density can also play a role in the recognition of similarities. More complex features such as harmony or structural features (for instance, how are the motives of the melody organized in time) should also be taken into account. Some studies have attempted to distinguish the features that are more relevant in our perception of similarity. Lamont and Dibben (2001) suggest that complex features and more simple features such as pitches or dynamics play different roles in the similarity judgments of listeners. Deliège (2001) insists on the fact that a training phase plays an important role for listeners in the recognizing of similarities between motives.

Before defining what is similar, a first step consists of selecting note sequences that could be possible patterns. For this task, a brute force matching of all patterns of all lengths would be too computationally demanding.

Rolland (1999) proposes a dynamic programming-based approach using structural, local and global descriptors for representing melodies. The algorithm first builds a graph of similarity from the musical sequences and then extracts patterns from the graph.

Another approach consists of filtering all the possible patterns of notes by considering musically advised grouping of melodic streams into phrases. Cooper and Meyer (1960) have proposed a set of rhythmic note grouping rules that could be used for this task. The rules are based on harmonic, dynamic, or melodic features (for instance, two pitches belonging to the same harmony or having the same frequency or duration will often be grouped). In the psychoacoustics area, McAdams (1994) has performed experiments that aimed at showing that notes could be perceptually grouped together according to various criteria (see also (Bregman, 1990) and (Smaragdis, 2001) for literature on perceptual grouping).

Other methods propose to optimize pattern finding. For instance, (Hsu et al., 1998), locate patterns using a correlative matrix approach. To make the process of finding repeating patterns more efficient, a matrix is used to keep the intermediate results of substring matching.

Once patterns have been located, similarity between patterns has to be quantified. We can find two different approaches when trying to measure similarity. These approaches are represented at figure 6.

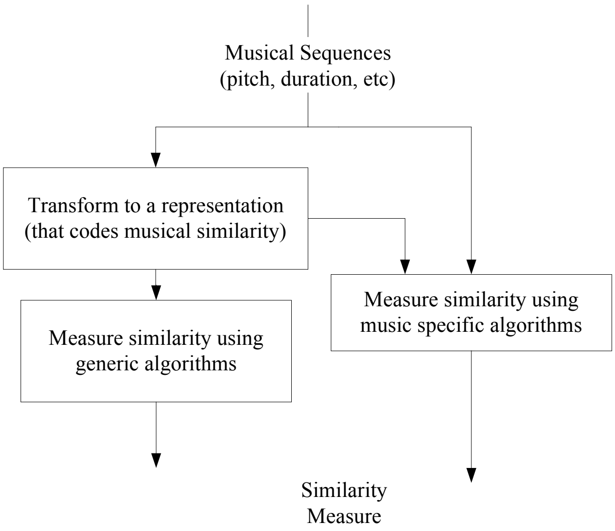


Figure 6: two approaches for measuring similarity between music sequences.

One elegant way to solve this problem is to describe the patterns in a way that musical pattern matching will turn into a generic (not music specific) pattern matching problem. Musical similarity is in that case implicitly coded into the melody representation (example with pitch contour is given below). Doing this, the musical specificities of the considered features will not have to be taken into account in the execution of the similarity measuring algorithms, and we will be able to use generic algorithms found in other scientific areas (as explained at section 5.1).

For instance, if the similarity is based on the pitch contour, then the pitch contour representation should be used to code the melody, if the similarity is based on the rhythm (which is invariant by scaling), then proportional representation should be used. One can also represent rhythm using a relative time base through the use of duration ratios instead of absolute time values (Stammen & Pennycook, 1993) (MPEG Working Documents, 2001). Duration ratios are calculated by dividing the duration of a note by the duration of the previous one. Finally, if the similarity is based on the invariance by transposition, inversion or melodic ornamentation, then a new representation format could be defined. A problem with the intervallic representation is that substituting one pitch value with another affects the intervals on both sides. Similarly, rhythmic disposition of one note destroys two duration ratios. This can be solved by calculating a cumulative sum over the intervallic representations in the course of the comparison. In this case, substitution or displacement of a single note affects only one interval (Rowe, 2001).

Assuming that the feature representations can be considered as numerical sequences abstracted from their musical context, usual similarity measuring algorithms can be used. The fact is that many algorithms that perform similarity measure can be found in other information retrieval areas than music, such as speech recognition (for instance the algorithm proposed in (Mongeau & Sankoff 1990)) or other areas such as mathematics and particularly statistics.

Uitdenbogerd and Zobel (1998) have tested several generic techniques for measuring the similarity: local alignment, longest common subsequence, an n-gram counting technique that include the frequency of occurrence of each n-gram, and the Ukkonen n-gram measure. The concluded that local alignment gives the best results. However it is a quite slow technique, thus faster n-gram based techniques were investigated. Results can be found in (Uitdenbogerd & Zobel 1999).

Another issue is how to consider several features. This poses the problem of how to evaluate the interaction of several dimensions on the similarity measure.

Two solutions have been proposed (see figure 7). One of them is to consider the dimensions separately, and then the global similarity could be calculated as a fusion of the several measures. However, the dimensions sometimes have to be considered simultaneously. For instance, two musical sequences could have a common melodic pattern on a defined period and a rhythmic pattern the rest of the time. Considering the two dimensions (pitch and rhythm) separately, the similarity would be quite low between the two sequences. Considering the two dimensions simultaneously, the similarity measure would be higher.

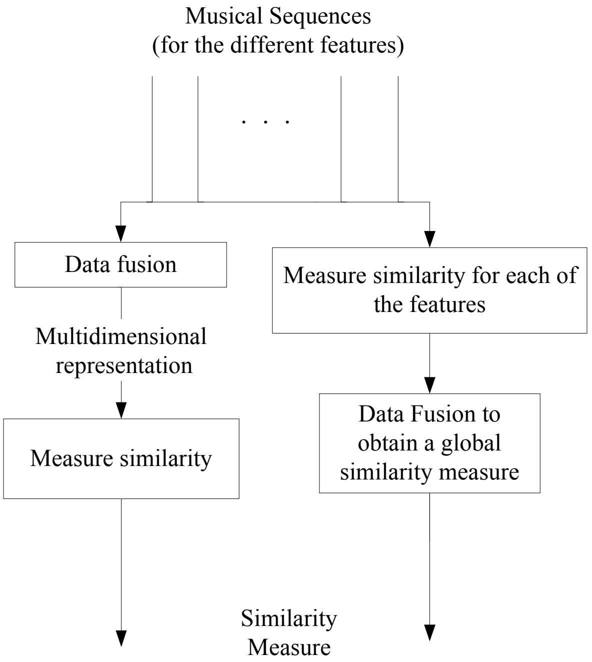

Musical Sequences
(for the different features)
. . .
Data fusion
Multidimensional representation
Measure similarity
Measure similarity for each of the features
Data Fusion to obtain a global similarity measure
Similarity Measure

Figure 7: Two approaches for taking into account several features when using generic similarity algorithms

Considering several features simultaneously, Li (2000) proposes to constitute a database of audio segment classes that will be used as reference for the similarity measure. Each class can be seen as a generalization of its components. Then, an input melody will be divided in segments that will be compared to the various classes of the database using the NFL (nearest feature line) algorithm. Li also evaluates the confidence in the similarity measure. In that way, all the features are considered simultaneously. However, specific high-level musical considerations, as invariance by contour, by scaling or by inversion, are not taken into account neither in the features representation by classes nor in the NFL algorithm. It could be interesting to apply this methodology to high-level musical representations too.

The second approach to measure similarity is to define specific algorithms adapted to musical specificities (for instance, an algorithm which states that two melodies, apparently different, are similar because they have the same structure *ababa*, can be said adapted to the structural specificity of music). If musically advised representations, which would turn musical pattern matching into a generic pattern-matching problem, are not used, the musical similarity information must be encoded to the matching algorithms.

There are some examples of this approach. Buteau and Mazzola (2000) propose a distance measure algorithm that takes into account a set of mathematical transformations on melodic patterns, as inversion or symmetric transformations. Clausen et al. (2000) and Crochemore et al. (2000) propose algorithms to measure the similarity between melodies containing note gaps. Cambouropulos (2000) (Cambouropoulos et al., 1999) and Rolland (1999) also propose algorithms being sensitive to approximate musical repetitions.

Damerau (1964) and Levenshtein (1966) have proposed an algorithm that measures a metric distance between two strings. It is based on several editing rules that transform one melody to the other. A cost can be associated to the transformations in order to give preference to some of them. Orpen and Huron (1992) notice that this algorithm is an approach to characterize the quantitative similarity for non-quantitative data (data which is not represented by numerical sequence, for instance a string-sequence). They have performed several experimentations of the algorithm (implemented at the *simil* command of the *Humdrum Toolkit* (Huron, 1993)), and he has concluded that it provides a promising way of characterizing the quantitative similarity between musical passages.

Dynamic programming has been successfully used to handle the cases of insertion, deletion, and substitution in melodic fragments. Dannenberg was among the first to apply dynamic programming to music. Together with Bloch, they later designed a program for real-time accompaniment of keyboard performances (Bloch and Dannenberg 1985). Computer accompaniment is closely related to score following that, in turn, requires pattern matching. The approach was later significantly developed by Stammen and Pennycook (1993) who included durational values for rhythm, dynamic time warping with global constraints, automatic segmentation into phrases through the implementation of the grouping rules proposed by Lerdahl and Jackendoff (1983). Subsequent improvements have included a real-time pattern induction and recognition approach of (Rowe, 2001) and inclusion of metrical stress and temporal weighting (O Maidin, 1998).

6 Melody transformations

Once we have a melody description scheme, we can define some transformation based on these melodic features. The needed transformations depend on the application. Amatriain et al. (2002) describe a set of operations for sound transformation, and presents some Artificial Intelligence based algorithms. High-level music transformations can be sorted in different groups:

Harmonic transformations

- *Modulation*: It consists on changing the tonality of the melody, thus requiring the knowledge of key and scale information.
- *Reduction*: It consists on simplifying the harmonic sequence (thus the melody) by cutting some harmonies, which do not play a given relevant role in the sequence. For instance, one could just keep the cadences and the modulations.
- *Harmonization*: The goal is to associate a chord sequence with a melody. Several chord sequences can be associated to a given melody, thus the harmonization can be different each time.

Depending on the melody, the last two tasks can be very difficult to be performed automatically at the signal level.

Melodic (horizontal) transformations

- *Transposition* (or pitch shifting): This consists on varying all the pitch values of a melody by a given interval. In a particular key, this could be done with the constraint of preserving the scale and, of course, relevant properties such as timbre and rhythm.
- *Symmetries*: It consists on performing axial or central symmetries to the melody. This corresponds to musical manipulations such as inversion, retrograde or contrary movement.
- *Ornamentation/reduction*: It consists on adding/deleting notes to the melody without changing its structure.

Rhythmic transformations

- *Time compression and dilatation* (time stretch): It consists on changing segment durations without modifying the rhythmic structure. Although it cannot be considered a rhythmic transformation (since it does not transform the rhythmic features) it relates to temporal aspects.
- *Various symmetries*: It consists on performing axial or central symmetries to the rhythmic sequence.
- *Accent and silences changes*: Changes on the accent and silences location also modifies the rhythmic features of an audio excerpt.

These transformations are usually performed on a midi-like music sequence, so that we assume that at least pitch and rhythm information are available. Some literature on musical transformation can be found in (Hammel, 2000).

Hofmann-Engl (2001) also formulates general melodic transformations such as transposition and inversion using reflection and translation matrices for measuring melodic similarity.

7 Conclusion

In this article we reviewed the main techniques dealing with melody in the context of music content processing. There is a substantial amount of research being currently carried out on this topic, and many methods are being studied to describe and process audio in a meaningful way.

In a broad melody processing system, different stages can be distinguished. The extraction of a melody description from the audio signal is one of the most important ones. We have reviewed some of the methods used for describing melodic features of an audio excerpt. In this phase, fundamental frequency estimation plays a major role.

If we think on sharing descriptions and communicating between applications and systems, it is necessary to define a common melody description scheme. This scheme ought to be valid for any application in any usage situation, which is a very hard requirement. It is more practical to try to devise a set of specific application contexts and define a description scheme for each of these target applications than considering an abstract, universal scheme.

MPEG-7 is the only standard that proposes an all-purpose melody description scheme. Some examples of MPEG-7 melodic description can be found in the MPEG schema specification documents (MPEG Working Documents, 2001) and (MPEG-7 Schema, 2001). This scheme provides two levels of description, the melody contour and the expanded description, as well as some information about lyric, scale and meter, as has been explained in section 3. This description scheme has been shown to present some limitations which we have divided in two groups.

First, comparing to a score representation, we can find that some important information is left out. For instance, as explained in section 3, melodies of very different nature can be represented similarly when using the melody contour. The expanded description also leaves out silences, which play an essential role in melodic perception.

Secondly, except for lyrics and phonemes, the description can be completely extracted from the score, without considering the audio signal. This means that some aspects of melody are left out by the description scheme, as information related to the performance (articulation, dynamics, deviation from the score, an other facets associated with expressivity) and some higher-level melodic aspects not derived from score information.

Some improvements could be proposed at different levels. For example, some of this signal-related information could be included as low-level descriptors (defined in (MPEG Working Documents, 2001)). Expanding the *mpeg7:Note* definition to add information about intra-note segments, articulation, and intensity could also incorporate other performance aspects. Another possibility would be to add to the melody description scheme some optional high-level melodic descriptors associated to the whole audio excerpt.

It is likely that when the MPEG-7 description scheme is adapted to use in different applications areas, certain number of additional needed improvements will arise.

Other important steps for melody processing are the techniques intended to locate melodic patterns or motives. Some of these techniques have been adapted from other disciplines, and some have been designed specifically for musical pattern processing. In both cases, the various specificities of music (temporal, structural, polyphonic and perceptive aspects) have to be taken into account, which makes those tasks difficult to achieve.

Finally, once we have a melody description at different levels, another important issue is how to transform the audio based on these melodic attributes. Techniques are currently under research, and they will give the possibility to transform the content using attributes at different levels: according to signal attributes, to musical attributes, textual attributes, etc.

There are many application contexts that need a melody description, and many of the techniques reviewed in this article have been designed having one of these different contexts in mind. One example is the *Query by Humming* systems, in which melodic representations are constructed and matched against a database of melodies. Another example arises in the field of managing music libraries, where it is necessary to derive melodic descriptions directly from audio files. Melody description has also been proved to be necessary when developing computer-assisted composition and improvisation tools and in the context of *Computational Auditory Scene Analysis* and *Music-Listening Systems*.

Acknowledgements

The authors would like to thank the members of the Music Technology Group at the Universitat Pompeu Fabra for their help and support.

The work reported in this article has been partially funded by the IST European project CUIDADO and by the TIC Spanish project TABASCO.

References

Amatriain, X., Bonada, J., Loscos, A., Arcos, J. L., Verfaille, V. (2002). Addressing the content level in audio and music transformations. In *Journal of New Music Research* (this issue).

Anderson, E. J. (1997). *Limitations of short-time Fourier transforms in polyphonic pitch recognition.* Ph.D. qualifying project report, Department of Computer Science and Engineering, University of Washington, Seattle, Washington, May 14, 1997.

Barthélemy, J. & Bonardi, A. (2001). Figured bass and tonalities recognition. In *Proceedings of the International Symposium on Music Information Retrieval*, Indiana University, Plymouth, Massachussets.

Blackburn, S. (2000). *Content based retrieval and navigation of music using melodic pitch contour.* PhD Thesis Report*,* University of Southampton.

Bloch, J. & Dannenberg, R. (1985). Real-time computer accompaniment of keyboard performances. In *Proceedings of the 1985 International Computer Music Conference.*

Bregman, A. S. (1998). Psychological data and computational auditory scene analysis. In *Computational auditory scene analysis*. Rosenthal, D and Okuno, H. G editors, Lawrence Erlbaum Associates, Inc., Mahwah, New Jersey.

Bregman, A. S. (1990). *Auditory Scene Analysis*. MIT, Cambridge, Massachusetts.

Buteau, C. & Mazzola, G. (2000). From contour similarity to motivic topologies. *Musicae Scientiae*, vol. 4, nº 2, pp. 125-149.

Cambouropoulos, E. (2000). Extracting 'significant' patterns from musical strings: some interesting problems. In *Proceedings of the London String Days workshop*. King's College London and City University.

Cambouropoulos, E., Chrochemore, M., Lliopoulos, S. C., Mouchard, L., & Pinzon, Y. J. (1999). Algorithms for computing approximate repetitions in musical sequences. *In Proceedings of the 10th Australasian Workshop On Combinatorial Algorithms*.

Cano, P. (1998). Fundamental frequency estimation in the SMS analysis. In *Proceedings of the 1998 Workshop on Digital Audio Effects*.

De Cheveigné, A. (2000). A note-lattice descriptor for melody. *MPEG-7 document number MPEG99/M6086.*

Clausen, M., Engelbrecht, R., Meyer D., & Schmitz, J. (2000). PROMS: A web-based tool for searching in polyphonic music. In *Proceedings of the International Symposium on Music Information Retrieval.*

Cooper, G. & Meyer, L. B. (1960). *The rhythmic structure of music*. University of Chicago Press.

Cope, D. (1998). *Signatures and earmarks: computer recognition of patterns in music*. MIT Press Cambridge, Massachusetts.

Crochemore, M., Iliopoulos, C. S., Pinzon, Y. J., & Rytter, W. (2000). Finding motifs with gaps. In *Proceedings of the International Symposium on Music Information Retrieval.*

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, vol. 7, nº 3, pp. 171-176.

Desain, P. & L. Windsor (editors) (2000). *Rhythm perception and production*. Lisse: Swets & Zeitlinger.

Deliège, I. (2001). Prototype effects in music listening: an empirical approach to the notion of imprint. In *Music Perception*, vol. 18, nº 3, Spring 2001.

Doval, B. & Rodet, X. (1991). Fundamental frequency estimation using a new harmonic matching method. In *Proceedings of the International Computer Music Conference*, pp. 555-558.

Doval, B. & Rodet, X. (1993). Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and HMMs. In *Proceedings of the IEEE-ICASSP*, pp. 221-224, 1993.

Educational Dictionary-Thesaurus (2000), Wordsmyth, <http://www.wordsmyth.net/>

Accepted for publication on May 9th 2002

Godsmark, D. & Brown, G. J. (1999). A blackboard architecture for computational auditory scene analysis. In *Speech Communication*, vol. 27, pp. 351-366, 1999.

Gold, B. & Rabiner, L. (1969). Parallel processing techniques for estimating pitch periods of speech in the time domain. *Journal of the Acoustic Society of America*, vol. 46, pp. 442-448.

Goto, M. (1999). A real-time music scene description system: detecting melody and bass lines in audio signals. In *Proceedings of the IJCAI Workshop on Computational Auditory Scene Analysis*.

Goto, M. (2000). A robust predominant-f0 estimation method for real-time detection of melody and bass lines in CD recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.* >

Gouyon, F., Meudic, B. (2002). Towards rhythmic content processing of musical signals: Fostering complementary approaches. *Journal of New Music Research* (in this issue).

Hammel, B. (2000). An essay on patterns in musical composition transformations, mathematical groups, and the nature of musical substance. web publication. <http://graham.main.nc.us/~bhammel/MUSIC/compose.html>

Hermansky, H., Morgan, N., & Hirsch, H. G. (1993). Recognition of speech in additive and convolutional noise based on RASTA spectral processing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 83-86.

Hess, W. (1983). *Pitch determination of speech signals. Algorithms and devices*. Springer-Verlag Berlin, New York, Tokyo.

Hofmann-Engl, L. (2001). Towards a cognitive model of melodic similarity. In *Proceedings of the 2nd International Symposium on Music Information Retrieval*. Indiana University, Plymouth, Massachussets, October 15-17.

Hsu, J. L., Chih-Chin, L., & Chen, A. L. P. (1998). Efficient repeating pattern finding in music databases. In *Proceedings of the 7th ACM International Conference on Information and Knowledge Management*. 3-7 November.

Huron, D. (1993). *The Humdrum Toolkit: Software for Music Research*. Center for Computer Assisted Research in the Humanities, Ohio State University, copyright 1993-1999.

Huron, D. (1997). Humdrum and Kern: selective feature encoding. In *Beyond MIDI: The handbook of musical codes*. Selfridge-Fiel1d, E editors, MIT Press, Cambridge, Massachusetts 1997.

Jehan, T. (1997). Musical signal parameter estimation. *CNMAT report*, 1997.

Kashino, K., Kinoshita, T., & Tanaka, H. (1995). Organization of hierarchical perceptual sounds: music scene analysis with autonomous processing modules and a quantitative information integration mechanism. In *Proceedings of the International Joint Conference On Artificial Intelligence*.

Kim, Y. E., Chai, W., Garcia, R., & Vercoe, B. (2000). Analysis of a contour-based representation for Melody. In *Proceedings of the International Symposium on Music Information Retrieval*, 2000.

Klapuri, A. (2000). Qualitative and quantitative aspects in the design of periodicity estimation algorithms. In *Proceedings of the European Signal Processing Conference*, 2000.

Klapuri, A., Virtanen, T., & Holm, J. M. (2000). Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals. In *Proceedings of the International Conference on Digital Audio Effects*, 2000.

Klapuri, A., Virtanen, T., Eronen, A., & Seppänen, J. (2001). Automatic transcription of musical recordings. In *Proceedings of the Consistent & Reliable Acoustic Cues Workshop*, 2001.

Lahat, A., Niederjohn, R. J., & Krubsack, D. A. (1987). A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech, In *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, nº 6, pp. 741-750.

Lamont A., Dibben, N. (2001). Motivic structure and the perception of similarity. In *Music Perception*, vol. 18, nº 3.

Laroche, J. (1995). Traitement des signaux audio-fréquences. *Ecole National Supérieure de Télécommunications report*.

Lartillot, O., Dubnov, S., Assayag, G., & Bejerano, G. (2001). Automatic modeling of musical style. In *Proceedings of the International Computer Music Conference*.

Lemström, K., & Laine, P. (1998). Musical information retrieval using musical parameters. In *Proceedings of the International Computer Music Conference*.

Lerdahl, F. & Jackendoff, R. (1983). *A generative theory of tonal music*. MIT Press, Cambridge, Massachusetts.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, vol. 10, pp. 707-710.

Li, S. Z. (2000). Content-based classification and retrieval of audio using the nearest feature line method. In *IEEE transactions on speech and content based processing*.

Lindsay, A. T. (1996). Using contour as a mid-level representation of melody. *Master of Science in Media Arts and Sciences Thesis*. Massachusetts Institute of Technology.

Lindsay, A. T. & Herre, J. (2001). MPEG-7 and MPEG-7 audio - an overview. In *Journal of the Audio Engineering Society*, vol. 49, nº 7/8, pp. 589-594.

Maher, R. C. & Beauchamp, J. W. (1993). Fundamental frequency estimation of musical signals using a two-way mismatch procedure. In *Journal of the Acoustic Society of America*, vol. 95, pp. 2254-2263, 1993.

Martin, K. D. (1996). Automatic transcription of simple polyphonic music: robust front end processing. *M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 399*, presented at the Third Joint Meeting of the Acoustical Societies of America and Japan, December 1996.

Mc Adams, S. (1994). Audition: physiologie, perception et cognition. In M. Richelle, J. Requin & M. Robert (eds.), *Traité de psychologie expérimentale*. Paris, Presses Universitaires de France, pp. 283-344, 1994.

Mc Nab, R. J., Smith, L. A., & Witten, I. H. (1996). Signal processing for melody transcription. In *Proceedings of the Australasian Computer Science Conference, pp. 301-307.*

Medan, J., Yair, E., & Chazan, D. (1991). Super resolution pitch determination of speech signals. *IEEE Transactions on Signal Processing*, vol. 39, nº 1, pp. 40-48.

Meddis, R. & Hewitt, M. J. (1991) Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification. In *Journal of the Acoustic Society of America*, vol. 89, nº 6, pp. 2866-2882, 1991.

Mongeau, M., Sankoff, D., (1990). Comparison of musical sequences. In *Computers and the Humanities*, vol. 24, pp. 161-175, 1990

MPEG-7 Schema (2001). Final Draft International Standard (FDIS) <http://pmedia.i2.ibm.com:8000/mpeg7/schema/>

MPEG Working Documents (2001). MPEG working group <http://www.cselt.it/mpeg/working_documents.htm>

Nettheim, N. (1992). On the spectral analysis of melody. In *Journal of New Music Research*, vol. 21, pp. 135-148.

Noll, A. M. (1967). Cepstrum pitch determination. In *Journal of the Acoustic Society of America*, vol. 41, pp. 293-309.

O Maidin, D. (1998). A Geometrical algorithm for melodic difference. In *Melodic similarity – Concepts, procedures, and applications*, Hewlett, W. B. & Selfridge-Field, E. editors, MIT Press, Cambridge, Massachussets, 1998.

Orpen, K. S. & Huron, D. (1992). Measurement of similarity in music: a quantitative approach for non-parametric representations. In *Computers in music research*, vol. 4, pp. 1-44.

Piszczalski, M. & Galler, B. A. (1979). Predicting musical pitch from component frequency ratios. In *Journal of the Acoustic Society of America*, vol 66, pp. 710-20.

Rabiner, L. R., Sambur, M. R., & Schmidt, C. E. (1975). Applications of a nonlinear smoothing algorithm to speech processing. In *IEEE Transactions ASSP*, vol. 23, nº 6.

Rabiner, L. R. & Schafer, R. W. (1978). *Digital processing of speech signals*. Prentice-Hall.

Ringer, A. L. (2002). Melody: Definition and origins. In *The New Grove Dictionary of Music Online,* Macy, L. Editor, Macmillan Online Publishing, <http://www.grovemusic.com>

Road, C. (1996). Pitch and rhythm recognition in MIDI systems. In *The Computer Music Tutorial*, The MIT Press Cambridge, Massachusetts, pp. 503-531.

Rolland, P.Y. & Ganascia, J. (1996). Automated motive-oriented analysis of musical corpuses: A jazz case study. In *Proceedings of the International Computer Music Conference*.

Rolland, P. Y. (1999) Discovering patterns in musical sequences. In *Journal of New Music Research*, pp. 334-350.

Romero, J. & Cerdá, S. (1997). Uso del análisis multirresolución para calcular el pitch de señales en presencia de ruido", Revista de Acústica (SEA), vol. 28.

Rowe, R. (2001). *Machine Musicianship*. MIT Press Cambridge, Massachusetts.

Scheirer, E. D. (2000). *Music listening systems*. PhD Thesis, Massachusetts Institute of Technology.

Selfridge-Field, E. (1998). Conceptual and representational issues in melodic comparison. In *Melodic Similarity – Concepts, Procedures, and Applications*, Hewlett,W.B. & Selfridge-Field, E. editors, MIT Press, Cambridge, Massachusetts.

Smaragdis, P. (2001). *Redundancy reduction for computational audition, a unifying approach*. PhD Thesis, Massachusetts Institute of Technology, Media Laboratory, May 2001.

Smith, L. A., Mc Nab, R. J., & Witten, I. H. (1998). Sequence-based melodic comparison: a dynamic programming approach. In *Melodic similarity – concepts, procedures, and applications*, Hewlett, W. B & Selfridge-Field, E editors, MIT Press, Cambridge, Massachussets, 1998.

Solomon, L. (1997). Music theory glossary. *Web publication*, last updated 2002. <http://solo1.home.mindspring.com/glossary.htm>

Stammen, D. & Pennycook, B. (1993). Real-time recognition of melodic fragments using the dynamic timewarp algorithm. In *Proceedings of the International Computer Music Conference*.

Talkin, D. (1995). Robust algorithm for pitch tracking. In *Speech Coding and Synthesis*, Kleijn, W. B and Paliwal, K. K editors, Elsevier Science B. V.

Terhardt, E. (1979). Calculating virtual pitch. In *Hearing Research*, vol. 1, pp. 155-182.

Terhardt, E., Stoll, G., & Seewann, M. (1981). Algorithm for extraction of pitch and pitch salience from complex tonal signals. In *Journal of the Acoustic Society of America*, vol. 71, pp. 679-688.

Uitdenbogerd, A. L. & Zobel, J. (1998). Manipulation of music for melody matching. In *Proceedings of the ACM Multimedia Conference*.

Uitdenbogerd, A.L. & Zobel, J. (1999). Melodic matching techniques for large music databases. In *Proceedings of the ACM International Multimedia Conference*.

Walmsley, P. J., Godsill, S. J., & Rayner, P. J. W. (1999). Bayesian graphical models for polyphonic pitch tracking. In *Proceedings of the Diderot Forum*, Vienna, December 1999.

Walmsley, P. J, Godsill, S. J, and Rayner, P. J. W. (1999). Polyphonic pitch tracking using joint bayesian estimation of multiple frame parameters. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.

Ziv, J. & Lempel, A. (1977). A universal algorithm for sequential data compression. In *IEEE Transactions on Information Theory*, pp. 337-343.

Accepted for publication on May 9th 2002