Towards Computational Auditory Scene Analysis: Melody Extraction from Polyphonic Music

Karin Dressler

Fraunhofer Institute for Digital Media Technology IDMT, Ilmenau, Germany
kadressler@gmail.com

Abstract. This paper describes an efficient method for the identification of the melody voice from the frame-wise updated magnitude and frequency values of tone objects. Most state of the art algorithms employ a probabilistic framework to find the best succession of melody tones. Often such methods fail, if there are several musical voices with a comparable strength in the audio mixture. In this paper, we present a computational method for auditory stream segregation that processes a variable number of simultaneous voices. Although no statistical model is implemented, probabilistic relationships that can be observed in melody tone sequences are exploited. The method is a further development of an algorithm which was successfully evaluated as part of a melody extraction system. While the current version does not improve the overall accuracy for some melody extraction data sets, it shows a superior performance for audio examples which have been assembled to show the effects of auditory streaming in human perception.

Keywords: computational auditory scene analysis, auditory stream segregation, melody extraction

1 Introduction

Melody is defined as a linear succession of musical tones which is perceived as a single entity. The melody is often the predominant voice in the sound mixture, this means it stands out from the background accompaniment. There are several features that increase the salience of the melody tone, for example loudness, frequency variation, timbre, and note onset rate. State of the art melody extraction algorithms mainly exploit two characteristics to identify the melody voice: 1) the predominance of the melody voice in terms of loudness and 2) the smoothness of the melody pitch contour.

At present two main algorithm types for the identification of the melody voice can be distinguished: on the one hand, probabilistic frameworks are used to find the optimal succession of tones. They combine pitch salience values and smoothness constraints in a cost function that is evaluated by optimal path finding methods like the hidden Markov Model (HMM) or dynamic programming (DP)

9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)
19-22 June 2012, Queen Mary University of London

methods. On the other hand, there are rule based approaches that trace multiple F0 contours over time using criteria like magnitude and pitch proximity in order to link salient pitch candidates of adjacent analysis frames. Subsequently, a melody line is formed from these tone-like pitch trajectories, using rules that take the necessary precautions to assure a smooth melody contour. Of course such a division is rather artificial. It is easy to imagine a system that uses tone trajectories as input for a probabilistic framework, and vice versa a statistical approach can be used to model tones. In fact, Ryynänen and Klapuri have implemented a method for the automatic detection of singing melodies in polyphonic music, where they derive a HMM for note events from fundamental frequencies, their saliences and an accent signal [1].

Most state of the art approaches use probabilistic frameworks that accomplish the tone trajectory forming and the identification of the melody voice simultaneously [2–4]. The application of a statistical model provides an out of the box solution that evaluates different features of the melody voice, as long as they can be expressed mathematically in a cost function or a maximum likelihood function.

Rao and Rao advocate dynamic programming over variants of partial and tone tracking, but also acknowledge the drawback of current statistical approaches [3]: While for rule-based methods alternative melody lines can be recovered quite easily, there is no effective way to retrieve alternative paths using the prevailing DP approach (i.e. the Viterbi algorithm), because the mathematical optimization of the method depends on the elimination of concurrent paths. Hence, it is not easy to state whether the most likely choice stands out from all other choices.

If there is a second voice with a comparable strength in the audio mixture, the identification of the predominant voice becomes a challenging problem. Of course, this assertion is also true for rule-based methods. Unfortunately, it is not unusual to find a strong second voice in real-world music, as a booming bass line is almost mandatory in many music genres. Masataka Goto describes a system for the automatic detection of the melody and bass line for real-world music in [5]. Using realistic assumptions about contemporary music, the problem of the concurrent melody and bass line is addressed by intentionally limiting the frequency range for both voices using band pass filters. Rao and Rao present an approach towards the solution of this problem in [3], giving an example for DP with dual fundamental frequency tracking. The system continuously tracks an ordered pair of two pitches, but it cannot ensure that the two contours will remain faithful to their respective sound sources.

Another problem to be addressed is the identification of non-voiced portions, i.e. frames where no melody tones occur. The simultaneous identification of the optimal path together with the identification of melody frames is not easy to accomplish within one statistical model, so often the voicing detection is performed by a separate processing step. Nonetheless, optimal path finding algorithms may be confused by rests in the tone sequence, especially because the usual transition probabilities do not apply in between melodic phrases.

An important characteristic of the human auditory system is the influence of note onset rate on the stream segregation. Tone sequences that are a quick succession of large intervals actually fail to form a recognizable melody, since the auditory system cannot integrate the individual tones into one auditory stream [6, chapter 2]. The integration or segregation of such a tone sequence depends markedly on the duration of the tones, so a voice processing algorithm should take into account such temporal aspects, too.

In this paper, we present an algorithm for the identification of the predominant voice in music that addresses some of the problems mentioned above. An auditory streaming model is implemented, which takes the frame-wise frequency and magnitude information of tones as input. With this information, so-called voice objects are established, which in turn capture salient tones close to their preferred frequency range. Although no statistical model is implemented, probabilistic relationships that can be observed in melody tone sequences are exploited. The presented method is a further development of an algorithm presented in [7]. The main technical difference over the baseline method is the renunciation of the mediated tone search using streaming agents. In the updated version, the voice object itself actively seeks the next voice tone. This is a big advantage, because – supplemental to increased algorithm performance – additional (voice dependent) search criteria can be integrated, like for example timbral features.

2 Statistical Properties of Melodies

By voices musicians mean a single line of sound, more or less continuous, that maintains a separate identity in a sound field or musical texture. The melody has certain characteristics that establish it as the predominant voice in the musical piece. Of course, a musical voice is not a succession of random notes – tones belonging to the same voice usually have a similar timbre, intervals between notes have a certain probability, there are rules regarding harmony and scale, and onset times of notes can be related to a rhythmical pattern.

Unfortunately the retrieval of high level musical features from polyphonic music is a challenging task in itself. Even for the most prominent voice (i.e. the melody), it is difficult to identify note onsets or to assign a note name to a tone with a varying frequency.

However, a melodic succession of tones has statistical properties that can be more easily exploited. Huron states that pitch proximity is the best generalization about melodies [8, chapter 5]. This statement is well supported by the interval statistics[1], as melodies consist mostly of tone sequences that are typically close to one another in pitch (see figure 1). Indeed, the unison is the most frequent interval by a great margin, followed by the whole tone interval.

[1] The Fraunhofer Institute in Ilmenau has gathered a collection of 6000 MIDI songs containing multiple genres, ranging from classical to contemporary charts music. Nearly one million notes were analyzed to compile a statistic of interval occurrences and the average note durations in melody tone sequences.

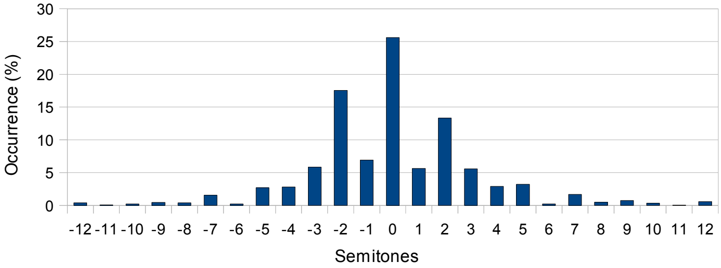


Fig. 1. Histogram of Note Intervals in Melodies

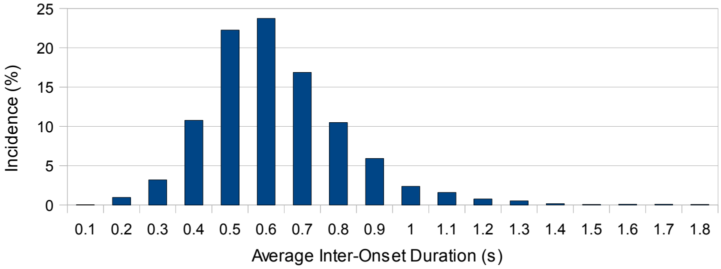


Fig. 2. Histogram of the Average Note Duration in Melodies

Other essential cues that help to distinguish musical voices are the central pitch tendency and regression to the mean [8, chapter 5]: the most frequently occurring pitches lie near the center of the melody's range. A necessary consequence of this tendency is the fact that after a melodic leap (an interval of more than three semitones) away from the center of the tone distribution, the following interval will change direction with a high probability. Regression to the mean is the most general explanation for this post-leap reversal.

The duration of melody tones lies normally in the range of 150 to 900 ms (see figure 2). Notes at faster rates occur, but they usually do not contribute to the perception of melody [9, chapter 5]. If a familiar tune is played at a rate faster than approximately 50 ms per note, the piece will not be recognizable, although the global melodic contour can be perceived. Yet, a very slow playback (i.e. durations of more than one second) is possible.

The dynamic range, which denotes the ratio between the largest and smallest occurring magnitudes in a tone sequence, is another important cue. Usually, tones that belong to the same voice have more or less the same sound level. It should be noted, however, that especially the human singing voice has a rather high dynamic range with ratios of more than 20 dB between the loudest tones and the softest ones.

The process that is required by the human auditory system as it analyzes mixtures of simultaneous and sequential sound entities has been coined auditory scene analysis [6]. All of the aforementioned statistical properties of melodies in fact enable the sequential grouping of sounds by the human auditory system. These "primitive" grouping principles are not only valid for music, but also for speech, environmental sounds, and even for noise.

Still, the ability of humans to distinguish concurrently sounding voices is limited. Huron investigates the ability of musically trained listeners to continuously report the perceived number of voices in a polyphonic musical performance in [10]. While Huron questions the musical significance of his experiment, because it does not evoke a natural listening situation, one important take away is that there is a marked worsening of the human performance, when a three-voice texture is augmented to four voices. If errors occur, the number of voices is underestimated in 92 percent of the cases. Another finding of the experiment is the fact that inner voices are more difficult to detect. The reaction time for the identification of an inner voice is twice as long, and often they are not detected at all.

3 Method

The formation of voices is controlled by the frame-wise updated magnitude and frequency of tone objects, which have a fundamental frequency in the range between 55 and 1319 Hz. The time advance between two successive analysis frames denotes 5.8 ms. Tone objects can be seen as pitch trajectories derived from salient pitches in so-called pitch spectrograms which may be computed with diverse pitch determination algorithms (PDA) like for example [11, 12]. Most PDA do not only compute pitch frequencies, but also offer an estimate for the corresponding pitch strengths, which is used as tone magnitude.

3.1 Overview

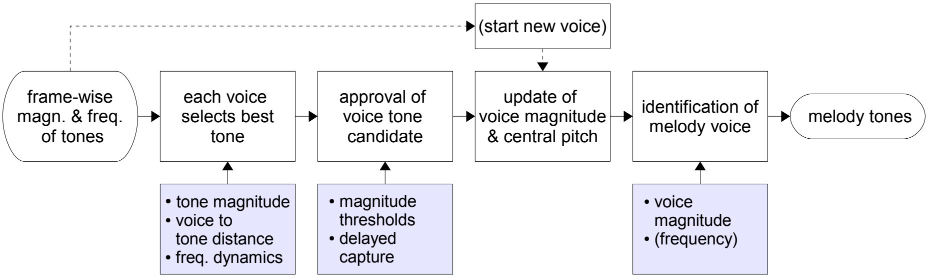

(start new voice)
frame-wise magn. & freq. of tones
each voice selects best tone
approval of voice tone candidate
update of voice magnitude & central pitch
identification of melody voice
melody tones
• tone magnitude
• voice to tone distance
• freq. dynamics
• magnitude thresholds
• delayed capture
• voice magnitude
• (frequency)

Fig. 3. Algorithm Overview

Figure 3 shows the processing steps performed in each analysis frame (i.e. every 5.8 ms). The input to the algorithm are the magnitude and frequency of the tone objects. The starting point of a new voice object is a salient tone which has not been added to an existing voice. In each analysis frame, every voice independently selects one tone, preferring strong tones that are close to its central pitch. If the selected tone passes all magnitude thresholds, it is added to the voice (after a certain delay period). The magnitude and central pitch of the voice are updated, whenever it has an added voice tone: the voice assembles a magnitude corresponding to the magnitude of the captured tone, and at the same time the voice's central pitch gradually moves towards the pitch of the added tone. Finally, the melody voice is chosen from the set of voices. The main criterion for the selection is the magnitude of the voice. Only tone objects of the melody voice qualify as melody tones.

3.2 Start Conditions

The first question to ask is at which point a new voice should be started[2]. The conditions for starting a new voice object are as follows:

- A voice is started from a tone which was not included in an existing voice.
- The tone reached at least once the maximum magnitude among all other tones.
- The magnitude of the tone has passed at least once the global magnitude threshold.
- There is no voice which could capture the tone, or the duration of the tone is greater than 200 ms, or the tone was finished.

3.3 Selection of Voice Tone Candidates

In each analysis frame, the voice object searches for a strong tone in the frequency range of ±1300 cent around its current central pitch. The best choice, at the one hand, ensures the smoothness of the voice tone sequence, at the other hand, embraces tones with a strong magnitude. In contrast to most existing approaches using optimal path finding methods, the smoothness of the melody line is evaluated in terms of central pitch, and not with respect to the last added tone. This strategy might not give the best results in every situation, but it reinforces the importance of the central pitch, and allows an easier recovery after an erroneous addition of a tone.

The Rating of Voice Tone Candidates: Each voice independently chooses only one tone – the object with the maximum rating A_{rating}:

$$A_{\text{rating}} = C \cdot D \cdot A_{\text{tone}} \cdot g_1(\varDelta c) \tag{1}$$

[2] The conditions given here are crafted for the purpose of melody extraction, which aims at the identification of the predominant melody line. If voices besides the predominant one shall be extracted, it is advisable to define more inclusive conditions.

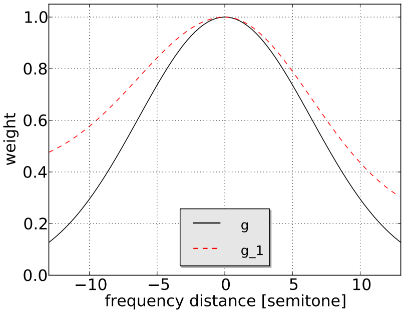


Fig. 4. Weighting Functions

The rating is calculated from the following four criteria:

- *Magnitude:* The tone magnitude A_{tone} is a good indicator for the perceptual importance of a tone.
- *Frequency distance weight:* The voice should preferably select a tone that is close to its central pitch. That is why the magnitude of the tone is weighted by a function that takes into account the frequency distance $\varDelta c$ between the tone's pitch c_{tone} and the central pitch of the voice $\bar{c}_{\text{voice}}$:

  $$g_1(\varDelta c) = r + (1 - r) \cdot g(\varDelta c) \qquad \text{with } \varDelta c = c_{\text{tone}} - \bar{c}_{\text{voice}}. \tag{2}$$

  The parameter $r = 0.4$ if $\varDelta c$ is negative, otherwise $r = 0.2$, and g is the function

  $$g(\varDelta c) = e^{-0.5 \frac{(\varDelta c)^2}{640^2}}. \tag{3}$$

  Figure 4 shows that the resulting weighting function g_1 is asymmetric – the weighting is biased towards tones from the lower frequency range. There are two reasons for this asymmetry. First, overtone errors cannot be avoided entirely, so in doubt the lower pitch is probably the true fundamental frequency. Second, tones in the lower frequency range of an instrument or the human voice are often softer, so the weighting compensates this difference.
- *Comparison with average magnitude:* The magnitude of the selected tone candidate should be in the order of the previously added magnitudes. For the comparison we use the maximum tone magnitude $\hat{A}_{\text{tone}}$ and the long term exponential moving average (EMA) of the maximum tone magnitudes of previously added tones[3]. If $\hat{A}_{\text{tone}}$ is more than 10 dB below or above the long term average, the rating is halved. Accordingly a magnitude factor C is set to 1 or 0.5 in the final rating.
- *Frequency deviation:* Sounds with changing attributes attract attention. Human listeners particularly focus on tones with vibrato or pitch glides. If a

[3] A detailed description of the exponential moving average can be found in the appendix.

tone shows persistently more than 20 cent frequency difference in between analysis frames the rating is doubled. Accordingly, a deviation factor D is set to 1 or 2 in the final rating.

Different Voices Competing for the Same Tone Any tone object preferably belongs to only one voice. In practice, there are often ambiguous situations, where an exclusive assignment to one voice is not the optimal solution.

The priority is on voices with a larger voice magnitude: a previously added tone may still be added by another voice, if the new owner has a larger magnitude than the current owner of the tone. Having said that, any voice which has a smaller magnitude than the current owner of the tone is prohibited to add the tone. The priority on strong voices is also reflected in the selection of the tones which was described in the previous section. The aim is that weaker voices shall avoid tones that are already added to strong voices (i.e. a voice with a large magnitude $\bar{A}_{\text{voice}}$). Hence, two more rating factors are introduced to direct the attention of weak voices to other suitable tone candidates:

- *Comparison of voice magnitude:* Whenever the tone is already included in a stronger voice, the original rating A_{rating} is multiplied with the factor 0.7.
- *Comparison of voice bidding:* If two voices aim at the same tone, the rating A_{rating} is decreased by the factor 0.7 for the voice with the lower bidding, but only if it is also the weaker voice. The voice bidding is the product of voice magnitude and the distance weight given in equation 2: $A_{\text{bidding}} = \bar{A}_{\text{voice}} \cdot g_1(\mathit{\Delta} c)$.

As the voice magnitudes and the voice biddings of the current frame are not known prior to the tone selection process, the values of the last analysis frame are used for the comparison. As the values usually change rather slowly, they are still significant. Furthermore, this provision ensures that the output is independent of the explicit order in which voices bid for tones.

3.4 Approval of Voice Tones

Even though one voice tone candidate is selected in each analysis frame, it is not clear whether the particular tone belongs to the voice or not, as melodies also contain rests. Two different techniques are employed to perform the voicing detection, namely the use of adaptive magnitude thresholds and the delayed capture of tones.

Short Term Magnitude Threshold The short term magnitude threshold is estimated for each voice individually. It secures that shortly after a tone is finished no weak tone is added to the voice prematurely. Hence, it is especially useful to bridge small gaps between tones of a voice. The short term threshold is adaptive and decays with a half-life time of 150 ms. Whenever the current voice tone has a magnitude which is larger than the current threshold reference value $T_{150\mathrm{ms}}$, it is updated to the new maximum:

$$T_{150\mathrm{ms}} \leftarrow \begin{cases} A_{\mathrm{tone}}, & \text{if} \quad A_{\mathrm{tone}} > T_{150\mathrm{ms}}; \\ \alpha_{150\mathrm{ms}} \cdot T_{150\mathrm{ms}}, & \text{otherwise.} \end{cases} \tag{4}$$

The parameter $\alpha_{150\mathrm{ms}}$ controls the decay of the magnitude threshold. The calculation of its value is described in equation 13. The tone passes the threshold if it is no more than 6 dB below $T_{150\mathrm{ms}}$.

Long Term Magnitude Threshold The long term magnitude threshold $T_{5\mathrm{s}}$ is basically the same as the short term threshold, with the distinction that it decays with a half-life period of 5 seconds. In order to pass the threshold, the tone's magnitude should not be more than 20 dB below $T_{5\mathrm{s}}$.

Long Term EMA Magnitude Threshold A high dynamic range of 20 dB within a tone sequence is not exceptional – a prominent example is the human singing voice. However, if a relatively high dynamic range is allowed, many tones from the accompaniment will pass the magnitude threshold, too. This is especially true for instrumental music, which often contains several simultaneous voices with a comparable strength. Besides the long term threshold, which is based on the maximum magnitude, another threshold is introduced which is computed as the exponential moving average of the previously added voice tone magnitudes. This threshold is updated whenever the voice has an approved voice tone, provided that the tone's duration is between 50 and 500 ms.

$$T_{\mathrm{EMA\_5s}} \leftarrow \alpha_{5\mathrm{s}} \cdot T_{\mathrm{EMA\_5s}} + (1 - \alpha_{5\mathrm{s}}) \cdot \hat{A}_{\mathrm{tone}} \tag{5}$$

The EMA is estimated with the current peak magnitude $\hat{A}_{\mathrm{tone}}$, which denotes the biggest magnitude the tone has reached so far. At the start of the voice the magnitude threshold is set to one third of the maximum magnitude of the first added voice tone. As the threshold reflects the dynamic range of previous voice tone magnitudes, the actual threshold value can be defined more strictly. In order to pass the threshold, the tone's magnitude should not be more than 10 dB below $T_{\mathrm{EMA\_5s}}$.

Delayed Capture of Tone The approval of a new voice tone is often delayed to allow some time for the start of a more suitable tone. The delay time depends on the distance between the candidate voice tone and the preferred frequency range of the voice (see section 3.5). All tones within the preferred frequency interval are added immediately, provided that they pass the magnitude thresholds. All other tones face a delay that depends on their magnitude and the frequency distance between tone and the preferred frequency range.

In order to estimate the delay, a short term pitch $\bar{c}_{\mathrm{st}}$ is defined for each voice object. (The computation of $\bar{c}_{\mathrm{st}}$ is described in section 3.5.) The tone may

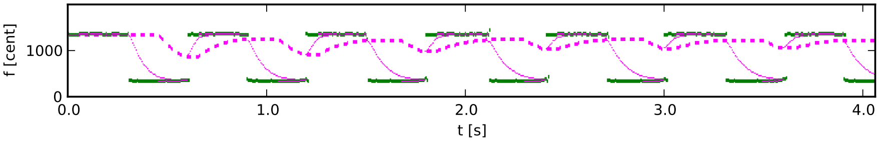


Fig. 5. Alternating tones: dashed line - central pitch of the voice $\bar{c}_{\text{voice}}$, thin line - short term pitch of the voice $\bar{c}_{\text{st}}$.

only be added after $\bar{c}_{\text{st}}$ has approximately reached the frequency of the voice tone candidate (i.e. less than 100 cent distance). Figure 5 illustrates the delayed capture of alternating tones.

3.5 Update of Voice Parameters

Contrary to the baseline method presented previously in [7], the intermediate step of streaming agents is omitted in this implementation. Consequently, voice objects do not derive their magnitude and central pitch from the assigned streaming agent. In the presented approach, the voice parameters are calculated directly based on the added tones.

Magnitude Update The voice magnitude $\bar{A}_{\text{voice}}$ is updated whenever the voice has an approved voice tone. The magnitude depends on the tone's rating magnitude A_{rating} as given in equation 1. The use of the rating magnitude ensures that a voice profits more from tones which are close to its current central pitch. In order to update the magnitude values, we use the exponential moving average (EMA).

$$\bar{A}_{\text{voice}} \leftarrow \alpha_{500\text{ms}} \cdot \bar{A}_{\text{voice}} + (1 - \alpha_{500\text{ms}}) \cdot A_{\text{rating}}. \tag{6}$$

The parameter $\alpha_{500\text{ms}}$ is a smoothing factor which corresponds to a half-life period of 500 ms. The EMA calculation is initialized with a fraction of the peak magnitude of the first tone: $\bar{A}_{\text{voice}} = 0.2 \cdot \hat{A}_{\text{tone}}$.

Central Pitch The central pitch of the voice $\bar{c}_{\text{voice}}$ is an important parameter, as it defines the preferred frequency range for the selection of tones. It is established over time according to the pitches of approved voice tones. While the adaptation could be implemented as EMA of previous frequencies, it is beneficial if the adaptation speed also depends on the tone's magnitude. This means the central pitch moves faster towards strong tones. That is the reason why at first a weight $\bar{A}_{\text{w}}$ is defined, which allows to evaluate the current rating of a tone in relation to the EMA of previous ratings:

$$\bar{A}_{\text{w}} \leftarrow \left(\bar{A}_{\text{w}} - A_{\text{rating}}\right) \alpha_{500\text{ms}} + A_{\text{rating}} \tag{7}$$

The EMA is initialized with $\bar{A}_{\mathrm{w}} = 0.2 \cdot \hat{A}_{\mathrm{tone}}$ at the beginning of the voice. With the help of the weight $\bar{A}_{\mathrm{w}}$ we can finally update the central pitch:

$$\bar{c}_{\mathrm{voice}} \leftarrow \frac{\bar{A}_{\mathrm{w}}\bar{c}_{\mathrm{voice}} + (1 - \alpha_{500\mathrm{ms}}) \cdot A_{\mathrm{rating}} \cdot c_{\mathrm{tone}}}{\bar{A}_{\mathrm{w}} + (1 - \alpha_{500\mathrm{ms}}) \cdot A_{\mathrm{rating}}}. \tag{8}$$

The parameter $\alpha_{500\mathrm{ms}}$ is a smoothing factor, which corresponds to a half-life period of 500 ms[4]. The parameter A_{rating} refers to the rating magnitude of the voice tone as given in equation 1, while c_{tone} is the pitch of the voice tone. The initial value for the iterative calculation is the frequency of the first added voice tone: $\bar{c}_{\mathrm{voice}} = c_{\mathrm{tone}}$. As $\bar{A}_{\mathrm{w}}$ is close to zero at the start of the voice, the central pitch changes more rapidly after the start of the voice (see also figure 5). This is, however, a deliberate decision, as the "true" central pitch has to be established over a longer time period.

Sometimes the frequency of a tone sequence does not prevail close to a central pitch, but moves upwards or downwards in one direction. As the central pitch adapts quite slowly, the update might not be fast enough to capture the succession of tones, and soon the tones fall outside the maximum search range of the voice. To avoid this, there is an immediate update of the central pitch, if $|\bar{c}_{\mathrm{voice}} - c_{\mathrm{tone}}| > 900$. In this case the central pitch is set to the maximum distance of 900 cent.

Frequency Range Intervals which are greater than an octave are rarely found in melody tone sequences. Consequently the search range for voice tones is limited to the range of ±1300 cent around the central pitch of a voice.

Moreover, a preferred frequency range R_{pref} is defined, which is given by the frequency range between the last added voice tone frequency and the central pitch of the voice.

Short Term Pitch The short term pitch $\bar{c}_{\mathrm{st}}$ seeks to emulate the time that is needed to focus attention to a tone that is outside the preferred frequency range of the voice. It is updated whenever the voice tries to capture a new voice tone, so it is updated even without an approved voice tone.

The short term pitch $\bar{c}_{\mathrm{st}}$ can immediately be set to any frequency within the preferred frequency range R_{pref}. So if the distance to a voice tone candidate can be decreased by changing the short term pitch to a frequency within R_{pref}, $\bar{c}_{\mathrm{st}}$ is set to that value. Apart from that, the short term pitch is updated very much like the central pitch of the voice – namely by using a weighted EMA. At first, a weight $A_{\mathrm{w\_st}}$ is defined, which allows to compare the tone's current rating A_{rating} with the magnitude of previously added voice tones. For this purpose we determine $A_{\mathrm{w\_st}}$ as the average of the long term EMA magnitude threshold

[4] Since the weight $\bar{A}_{\mathrm{w}}$ depends on many factors, the parameter α does not exactly set any half-life period for the central pitch update. Yet the corresponding time span gives a reference point for the approximate adaptation speed.

$T_{\mathrm{EMA\_5s}}$ and the short term magnitude threshold T_{150ms}:

$$A_{\mathrm{w\_st}} = 0.5 \cdot (T_{\mathrm{EMA\_5s}} + T_{\mathrm{150ms}}). \tag{9}$$

Finally, the short term pitch is updated using a weighted EMA:

$$\bar{c}_{\mathrm{st}} \leftarrow \frac{A_{\mathrm{w\_st}}\bar{c}_{\mathrm{st}} + (1-\alpha_{\mathrm{30ms}}) \cdot A_{\mathrm{rating}} \cdot c_{\mathrm{tone}}}{A_{\mathrm{w\_st}} + (1-\alpha_{\mathrm{30ms}}) \cdot A_{\mathrm{rating}}}. \tag{10}$$

The parameter α_{30ms} is again the smoothing factor. Figure 5 shows how $\bar{c}_{\mathrm{st}}$ is used to capture tones: only if the thin line reaches the voice tone candidate (i.e. less than 100 cent distance), the tone may be added to the voice.

3.6 The Identification of the Melody Voice

The most promising feature to distinguish melody tones from all other sounds is the magnitude. The magnitude of the tones is of course reflected by the voice magnitude. Hence, the voice with the highest magnitude is in general selected as the melody voice. It may happen that two or more voices have about the same magnitude and thus no clear decision can be taken. In this case, the voices are weighted according to their frequency: voices in very low frequency regions receive a lower weight. The magnitude thresholds are defined for each voice individually. As they depend solely on the past tones of the voice, they cannot take effect on all soft tones. Therefore, it is recommended that a global magnitude threshold is estimated from the identified melody tones. Subsequently, the melody tones should be compared to the global threshold.

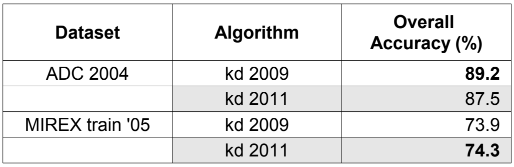
4 Results

4.1 Audio Melody Extraction

The presented method for the identification of musical voices has been implemented as part of a melody extraction algorithm which was evaluated using the melody extraction training data sets of ISMIR 2004 and MIREX 2005. Algorithm parameters regarding the width and the shape of the weighting functions as well as the timing constants of the adaptive thresholds have been adjusted using the same data sets. The previous algorithm version, which has been described in [7], is used as a benchmark[5]. The comparison with the previous algorithm version (kd2009) shows that the overall accuracy is not improved by the new method (kd2011)(see table 1). However, the results of the previous algorithm should not be seen as a baseline, as it still can be considered as a state of the art algorithm. Table 2 shows that its overall melody extraction accuracy is close to the best algorithm of the most recent MIREX audio melody extraction task, which was submitted by Salamon and Gómez [14].

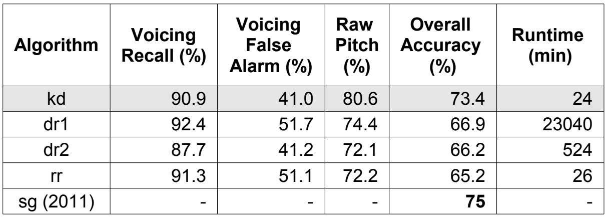
[5] The melody extraction algorithm using the previous voice detection method was evaluated at the Music Information Retrieval Evaluation eXchange (MIREX) [13].

Table 1. Comparison of Melody Extraction Results for the Training Datasets



Dataset	Algorithm	Overall Accuracy (%)
ADC 2004	kd 2009	**89.2**
	kd 2011	87.5
MIREX train '05	kd 2009	73.9
	kd 2011	**74.3**

Table 2. Melody Extraction Results of MIREX 2009 (4 best submissions and the best submission of MIREX 2011)



Algorithm	Voicing Recall (%)	Voicing False Alarm (%)	Raw Pitch (%)	Overall Accuracy (%)	Runtime (min)
kd	90.9	41.0	80.6	73.4	24
dr1	92.4	51.7	74.4	66.9	23040
dr2	87.7	41.2	72.1	66.2	524
rr	91.3	51.1	72.2	65.2	26
sg (2011)	-	-	-	**75**	-

One problem of the evaluation is that a melody extraction system is only interested in the predominant voice. The previous algorithm, as well as optimal path finding algorithms, already gives satisfactory results as long as the melody voice is indeed predominant. If the audio signal contains concurrent voices of comparable strength, it is important that all strong voices are retrieved, so that the final decision can be based on a more complete picture of the audio input. A qualitative comparison of the algorithm outputs allows a more meaningful evaluation than numbers alone.

4.2 Qualitative Analysis

A qualitative analysis of the results confirms that the new method has indeed some advantages over the baseline method:

- The minimum distance between two voices is decreased. (see figures 6 - 8)
- The detection of weak voices is improved. (see figures 6 - 8)
- The behavior of the algorithm is closer to human perception, when artificial audio examples for auditory stream segregation are used as input. (see for example figures 5 and 9)
- The implementation of the new method is more straight forward, because the intermediate processing of so-called streaming agents, was omitted (see reference [7]).
- The proposed method allows a simpler inclusion of timbral features, as the voice tone candidates are directly selected by the voice and not by streaming agents.
- The computation time for the voice processing scales with the complexity of the audio input.

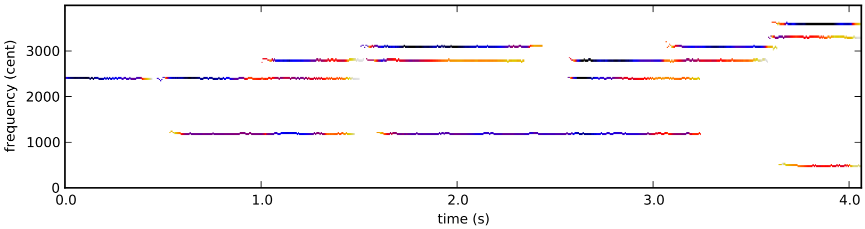


Fig. 6. Midi3.wav from the ADC 2004 data set is an example for instrumental music with several concurrent voices. The figure shows the identified tone objects, which constitute the input to the voice processing algorithm. The melody voice is in the high frequency range. The melody voice is the predominant voice, but the bass voice has a comparable strength.

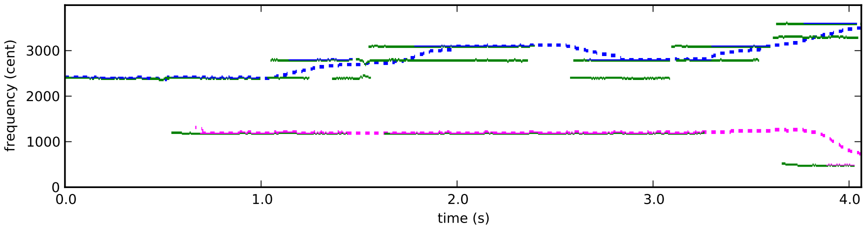


Fig. 7. Baseline method: When the bass voice starts, a second voice object is created. Two voices are recognized – the melody voice (blue) and the bass voice (magenta).

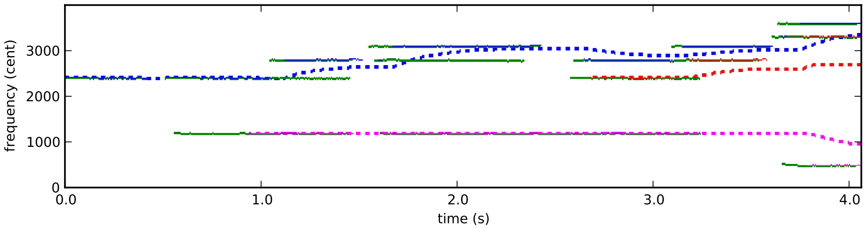


Fig. 8. Proposed method: Three voices are recognized – the melody voice (blue), the bass voice (magenta), and an inner voice (red).

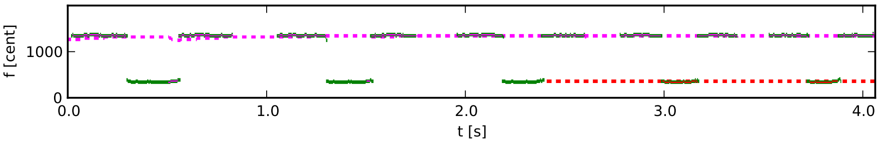


Fig. 9. Example of alternating tones: The duration of the tones is decreased over time. Soon the alternating tones cannot be captured by the first voice and a second voice is started.

5 Summary and Conclusion

In this paper, we presented an efficient approach to auditory stream segregation for melody extraction algorithms. The proposed method allows a reliable identification of a variable number of simultaneous voices in different kinds of polyphonic music. The qualitative comparison with a previous implementation shows that the proposed method improves the detection of musical voices. Furthermore, the new approach offers more possibilities to add voice dependent features for the tone selection in future implementations. Taking into account not only the magnitude and the occurring frequency intervals, but also the duration of tones, the presented algorithm is another step towards auditory stream segregation as performed by the human auditory system.

6 Appendix

6.1 Exponential Moving Average

A simple moving average is the mean of the previous N data points. An exponential moving average (EMA) applies weighting factors to all previous data points which decrease exponentially, giving more importance to recent observations while still not discarding older observations entirely. The smoothing factor α determines the impact of past events on the actual EMA. It is a number between 0 and 1. A lower smoothing factor discards older results faster.

The computation of the EMA can be expressed by the following formula

$$\bar{y}_l = (1 - \alpha) \sum_{i=0}^{l-1} \alpha^i y_{l-i}, \tag{11}$$

where l designates the current time period (i.e. current analysis frame), y_l is the current observation, and $\bar{y}_l$ the resulting EMA value.

However, the application of equation 11 is inconvenient, because all previous data samples have to be weighted and summed in order to compute the EMA. The same result can be achieved using the following recursive formula for time periods $l > 0$:

$$\bar{y}_l = \alpha \cdot \bar{y}_{l-1} + (1 - \alpha) \cdot y_l. \tag{12}$$

Equation 12 shows that the EMA can be calculated very efficiently from only two numbers: the current observation data y_l and the preceding EMA value $\bar{y}_{l-1}$. Thus, a big advantage of this method is that no previous data has to be stored in memory (besides the last EMA value).

In order to make the first recursive computation possible, the EMA value has to be initialized. This may happen in a number of different ways. Most commonly $\bar{y}_0$ is initialized with the value of the first observation. The problem of this technique is that the first observation gains a huge impact on later EMA results. As another option, the first EMA value can be set to 0. In this case the observations have comparable weights, but the calculated EMA values do not represent an average of the observations. Rather, the EMA value starts close to zero and then approaches the average slowly – just like a sampled capacitor charging curve.

For the actual implementation it is important to figure out optimal values for the smoothing factor α. A more intuitive measure than the smoothing factor is the so-called half-life period. It denotes the time span needed to decrease the initial impact of an observation by a factor of two. Taking into account the desired half-life t_h and the time period between two EMA calculations Δt, the corresponding smoothing factor is calculated as follows:

$$\alpha = 0.5^{\frac{\Delta t}{t_h}} . \tag{13}$$

References

1. M. Ryynänen and A. Klapuri. Transcription of the singing melody in polyphonic music. In *Proc. of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, Victoria, Canada, Oct. 2006.
2. C.-L. Hsu, L.-Y. Chen, J.-S. R. Jang, and H.-J. Li. Singing pitch extraction from monaural polyphonic songs by contextual audio modeling and singing harmonic enhancement. In *Proc. of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, Oct. 2009.
3. V. Rao and P. Rao. Improving polyphonic melody extraction by dynamic programming based dual f0 tracking. In *Proc. of the 12th International Conference on Digital Audio Effects (DAFx)*, Como, Italy, Sept. 2009.
4. J.-L. Durrieu, G. Richard, B. David and C. Fvotte Source/Filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech and Language Processing*, 18(3):564–575, Mar. 2010.
5. M. Goto. A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication (ISCA Journal)*, 311–329, Sept. 2004.
6. A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*, volume 1 MIT Press paperback. MIT Press, Cambridge, Mass., Sept. 1994.
7. K. Dressler An auditory streaming approach for melody extraction from polyphonic music. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, Miami, Florida, Oct. 2011.
8. D. Huron. *Sweet Anticipation: Music and the Psychology of Expectation.* The MIT Press, Cambridge, Massachusetts, 2006.
9. R. M. Warren. *Auditory perception: an new analysis and synthesis.* Cambride University Press, 1999.
10. D. Huron. Voice denumerability in polyphonic music of homogeneous timbres. *Music Perception*, 6(4):361–382, 1989.
11. K. Dressler Pitch estimation by the pair-wise evaluation of spectral peaks. In *AES 42nd Conference*, Ilmenau, Germany, July 2011.
12. D.J. Hermes Measurement of pitch by subharmonic summation *Journal of the Acoustical Society of America*, 83(1):257–264, 1988.
13. K. Dressler Audio Melody Extraction for MIREX 2009. In *5th Music Information Retrieval Evaluation eXchange (MIREX)*, 2009.
14. J. Salamon and E. Gómez Melody extraction from polyphonic music: MIREX 2011. In *7th Music Information Retrieval Evaluation eXchange (MIREX)*, 2011.