

A Hierarchical Approach to Onset Detection

Emir Kapanci and Avi Pfeffer

Division of Engineering and Applied Sciences, Harvard University
{kapanci,avi}@eecs.harvard.edu

Abstract

Onset detection in vocal music and many other instruments is complicated by the possibility of soft transitions between notes. Most systems try to identify onsets within a short-time window as it is easier to define transition functions over a restricted space. However, it may not be possible to detect soft onsets without considering a long-time window, for which defining and computing the transition function can be hard and computationally costly. We present a method which looks for onsets between locations of increasing distance and is able to capture such onsets without considering all the points within the window. For the onset identification function we use both a simple manual function and support vector machines trained using a labelled corpus.

1 Introduction

Onset detection is the problem of segmenting an acoustic signal into discrete events such as individual notes or rests. It is an important component of the analysis of acoustic signals, automated accompaniment, audio/video synchronization, audio editing and music transcription. In onset detection¹, the audio signal is usually viewed as a succession of very short frames (e.g. 10 ms) and the task is to identify locations where the frames on the two opposite sides differ significantly.

We use the term 'smooth instrument' for instruments that are capable of changing the pitch without a sharp change in the amplitude (e.g. bowed strings and voice). Such smooth transitions between two notes will be denoted as 'soft' onsets, while onsets accompanied with a sharp change in amplitude (including a note beginning after a rest, or a rest after a note) are denoted as 'hard' onsets. Soft onsets are much harder to identify, as they can be confused with vibrato or sharpening/flattening of pitch.

The traditional approach to onset detection is to look at the amplitude difference between successive frames and label those that are above a threshold as onsets. Another approach is to detect steady-state regions and then look backwards for sharp changes in amplitude (Foster, Schloss, and Rockmore

1982). However, for smooth instruments there may not be a sharp change of amplitude between notes, and soft onsets will be missed. Other systems split the signal into frequency bands and look for onsets in these instead of the signal as a whole (Scheirer 1998; Klapuri 1999). The outputs from separate bands then need to be combined to identify global onsets, using one of the many proposed methods, including neural networks (Marolt, Kavcic, and Privosnik 2002). It is also possible to use completely different techniques in separate bands, such as an amplitude based detection in upper bands and a frequency based detection in lower bands (Duxbury, Sandler, and Davies 2002).

Clearly, to identify soft onsets features other than the amplitude must be included, as the perception of onset can also result from other factors, such as a variation of pitch alone in legato violin playing, or a variation in timbre alone due to a change of vowel in vocal music. Various features can be used together to measure the difference between frames. For instance, an amplitude based approach can be combined with tonal analysis by looking at the deviations in expected phase (Duxbury et al. 2003). Even a more complete signal representation such as a spectrogram can be used to compute the difference between frames (Hainsworth and Macleod 2003).

While the addition of features is beneficial, it is not enough to solve the problem of soft onsets. Unlike hard onsets, soft onsets take some time to be established. The individual differences between successive frames are usually not large enough to be perceived as an onset, and it might take a large sequence of such small changes to yield one. If an onset detection algorithm uses a difference function that does not cover the entire soft onset region, it might be missed. Trying to accommodate this by allowing smaller changes to be recognized as onsets might introduce false positives. It can be very difficult to design functions over a sequence of frames of arbitrary length without introducing false positives or false negatives. In this paper, we recast the problem of onset detection into that of deciding whether two frames belong to the same event. First, we present a hierarchical method that looks for the difference between pairs of frames of increasing temporal distance and identifies onsets without considering all the points within the window. We then evaluate our algorithm on a corpus of vocal music, which poses the additional challenge of vowel changes. We conclude with discussion and future directions.

¹For the rest of the paper 'onset' denotes the beginning of a new note or a rest, i.e. we do not distinguish between note onsets and offsets.

2 Approach

To capture soft onsets ranging over an unknown number of frames, we suggest posing the problem of onset detection as that of deciding whether two frames of a certain temporal distance could be coming from the same event. A negative answer suggests the existence of an onset somewhere between the two frames. The motivation is that even if the onset is extremely smooth, for the perception of an onset to occur, there must exist a difference in either amplitude, pitch or timbre. However, the difference might only become perceivable over a sequence of frames, and the length of this sequence is highly variable even in a fixed instrument. The solution is to design an algorithm that does not use a comparison function over frames in a fixed-sized window, but an adaptive one. When we rephrase the problem as above, there is a clear function: it takes as inputs the features and the temporal distance of two frames, and returns +1 if there should be an onset between the two and -1 otherwise.

Although an onset implies a difference in one or more of the features listed above, the converse is not true. In many instruments there can be a significant change in these features within a single note event. Singers for instance, tend to change the amplitude of the sound particularly near the beginning and ending of a note. The pitch might also become sharper or flatter and the vowel quality can change dramatically over the duration of a note. Expressive singing magnifies all these effects. To avoid marking such changes as onsets, we should avoid comparing frames that are significantly far, or preferably the frame comparison function should be able to recognize that the further two frames are the more different they can be even though they belong to the same event.

In our approach, to locate onsets as precisely as possible, we start by comparing adjacent frames and increase the distance until reaching a threshold above which changes are little indicative of an onset. In doing so, we also want to avoid performing comparisons between all possible pairs, so we use the hierarchical structure explained in section 2.1 to select and order the comparisons that take place.

2.1 Comparison Graph

We use a directed graph to determine the comparisons for the onset detection. Each node in this graph represents a sequence of frames, and is labelled with the first and last frames of that sequence. Each node has an attribute indicating whether there is an onset within that region. The onset detection algorithm involves computing the values of these attributes. The structure of the tree determines how each region is divided into subregions, represented by child nodes. The detection algorithm first tries to identify onsets within smaller regions, then considers the union of the regions. So, the children are always processed before their parents. Figure 1 illustrates a comparison graph for an audio segment of 16

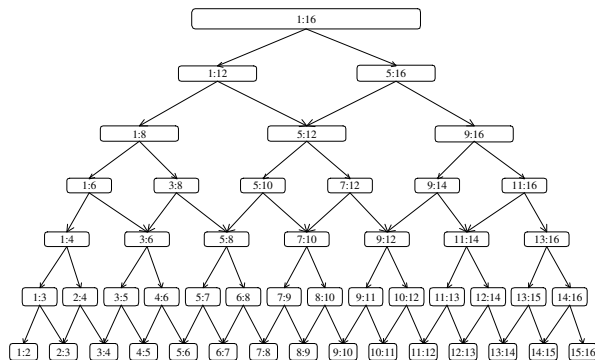


Figure 1: Comparison Graph

frames. Here, each region is divided into two overlapping subregions. The tree is built up by alternating levels where nodes are either $2/3$ or $3/4$ of the width of their parent. This configuration results in comparison of all frames of distance 1 or 2, as well as having the property that the grandchildren of each node represent various subsequences half the size of the sequence represented by that node. So, the width of the regions is halved (i.e. the precision is doubled) every two levels. Although we believe this is an adequate configuration, it can be modified to yield a finer/coarser onset detection. For instance, a faster -but less precise- detection can be obtained by halving the region width at every level.

The graph is processed bottom-up starting from the leaf nodes. The leaves represent two adjacent frames, so we simply compare these two frames and mark the node accordingly: +1 if there is an onset between them, -1 if not. At each non-leaf node, we first see if any of its children contain an onset. If that is the case, we can automatically mark that node as +1 without any comparison. Otherwise, we compare the two frames indicated by the node's label, and mark it accordingly. Once the root of the graph is processed all the nodes will have been marked. At that point, we can walk the graph to return the detected onsets: the label of any node marked +1 that has all its children marked -1. This ensures that the algorithm returns the smallest region it can restrict every onset to. An onset detected at a leaf node is a hard onset, while one detected at a non-leaf nodes is a soft onset.

2.2 Onset Recognition Function

To complete the algorithm, we need to design or learn a function that inputs the features and the temporal distance of a pair of frames and decides whether there must be an onset between the two. The features we use from each frame are the fundamental frequency, the amplitude, and relative strengths of first three harmonics. The fundamental frequency is determined using a simple frequency-domain method, where we pick the lowest frequency whose amplitude is above a threshold (similar to de Cheveigne and Kawahara (2002)). To deal

with some vowels where the second harmonic can be significantly stronger than the fundamental, we also look at half of the peak frequency and check whether it is above a second threshold. This results in smoother pitch tracks, with almost no octave errors. These features, as well as the distance between the frames are the inputs to the onset recognition function. We suggest using support vector machines to decide whether two frames could be resulting from the same event. To emphasize the contribution of the comparison graph we also show the results with a very simple manual function.

Support Vector Machines (SVM) SVMs are powerful learning machines for binary classification. The main idea is to use a kernel function to map the input features into a higher dimension and then learn an optimal hyperplane that separates the positive and negative examples with a maximal margin (see Cristianini and Shawe-Taylor (2000) for an introduction to SVMs). We used the LIBSVM library (Chang and Lin 2001) for the implementation of the SVMs in our learning. Each training example consisted of the features of a pair of frames, their distance, and a label indicating whether there is an onset between the two. The training examples were extracted from the corpus by taking all non-silent pairs of frames of distance 2^i for $i = 0$ to 4. As there were fewer positive examples in the training set, they were assigned a weight 10 times larger than that of the negative examples. Frames further than 160ms apart were not used in the training: this is the threshold above which the comparison function automatically returns -1. For the SVM, we used a Gaussian RBF kernel and the parameters were optimized by cross-validation on the training set using a grid search. Once a binary classifier is learned offline, given a pair of frames it returns the most likely class: +1 for onset, -1 for no onset.

Manual The manual function simply looks at the difference between the fundamental frequencies of the two frames and returns +1 (i.e. there must be an onset) if the difference is larger than a half-tone, and -1 if it is not.

2.3 Non-Hierarchical Approach

We also implemented a non-hierarchical algorithm that compares frames over long distances. The input features are the same, but instead of comparing only the frames at the ends of a region, the classification function inputs all the frames in that region and returns whether the central frame is part of an onset. Then to detect the onsets, we linearly go through each frame in the audio sequence, input the the region around them to the classification function and depending on the output mark them as belonging to an onset or not. We again use support vector machines to learn this function. As we allow more neighboring frames, the prediction accuracy increases at the cost of increased learning and prediction times. Since

the comparison upperbound for the hierarchical approach was 160ms, we used a neighborhood of 8 frames on both sides of each frame under consideration to ensure that we are able to cover regions up to the same size.

3 Evaluation

3.1 Training Data

To evaluate our approach, we assembled a manually labelled corpus of solo singing containing 15 song segments by Palestrina (of average length 26sec). The singer was a music student with vocal training, and was instructed to sing naturally with lyrics. As the main focus of the work was to identify soft onsets, monophonic music was chosen to ensure the accuracy of the hand-labelling. The choice of vocal music also introduces a challenging type of soft onset: changes in vowel without a change in pitch or amplitude.

For labelling, the sound files were split into 10ms frames. The hard onsets were marked between the two adjacent frames where they occurred. The soft onsets were divided into separate cases: a smooth upwards or downwards change in *pitch*, and a change only in the sung *vowel*. The pitch changes were marked with the help of the pitch track, and the vowel changes were marked with the help of a graph showing relative strengths of the harmonics. The resulting soft onset lengths are between 30–120ms. There are a total of 541 onsets in the corpus, 142 of which are soft.

3.2 Experimental Results

A commonly used metric for the evaluation of onset detection is $Accuracy = \frac{T - FP - FN}{T} \times 100$, where T is the total number of true onsets, FP is the number of extraneous onsets, and FN is the number of onsets missed (Klapuri 1999; Duxbury, Sandler, and Davies 2002). We also report the *Precision* and *Recall* over all onsets, as well as the recall for soft onsets and those due to vowel changes alone, as these have clearer interpretations. The precision is the percentage of the onsets the algorithm returns that are correct; and the recall is the percentage of the true onsets that were found. Both SVM results shown in Table 1 were obtained using averages from 5-fold cross-validation: the available data was split into 5 sets and each set was held-out in turn for testing, with the 4 remaining sets used for training. For the manual function all available data was used for testing, as no learning is needed.

The hierarchical approach using the SVM classifier outperforms the non-hierarchical approach in precision, accuracy and F-measure. It also has a higher recall of soft onsets, and in particular, all the onsets due to a vowel change alone are detected. Furthermore, it learns a simpler classification function: it uses 11 features while the non-hierarchical approach uses 85. As a result, both learning and onset detection

Table 1: Results

	Precision	Recall	Soft Recall	Vowel Recall	Accuracy	F-measure ²
Non-Hierarchical w/ SVM	82	94	80	83	75	87
Hierarchical w/ SVM	87	93	84	100	80	90
Hierarchical w/ manual	83	95	92	65	79	89

times are longer for the non-hierarchical approach.

The advantage of the comparison graph is particularly evident in the 3rd row of Table 1. Although it is much faster, the hierarchical approach with the manual function outperforms the non-hierarchical approach in everything other than recall of vowel changes. As it only has one feature (frequency difference) that might not capture timbral changes, a low recall of vowel changes was expected. However, it was able to identify an impressive 92 percent of all soft onsets. This indicates that by using the comparison graph, the onset detection function no longer has to be very complex. Since we get more than one shot at the onset candidates, each time using different information, we can postpone the decision of an onset until it becomes more evident.

4 Conclusion and Future Work

Detecting soft onsets, particularly those due to vowel changes, is a difficult task. By using a set of appropriate features over sufficiently large distances, we were able to detect most of the soft onsets in a corpus of vocal music, using both a hierarchical and a non-hierarchical approach. The features from each frame included the frequency, amplitude and relative strength of harmonics.

The non-hierarchical approach looked at continuous sequences of frames. Our results show that the features provided sufficient information to detect most onsets. However, this approach is computationally costly since covering regions long enough to detect soft onsets requires looking at the features of a large number of frames.

In our hierarchical approach, we posed the problem of onset detection as asking whether two frames could be coming from the same event. We introduced a graph structure which selects and guides the order of the comparisons. Closer frames are compared first to detect onsets as precisely as possible. To capture soft onsets, frames of increasingly longer distance are compared. Since we no longer consider all the frames in a sequence but only the end points, independent of the size of the region where we look for onsets, we always have two frames that are input to the classification function. Our results show that this is an effective way to formulate the onset detection problem: the hierarchical system was able to detect most of the soft onsets even when using a very sim-

ple classification function. Furthermore, our hierarchical approach using support vector machines was very successful in detecting onsets due to vowel changes, which is a challenging type of onset.

The technique proposed in this paper can be used in conjunction with most of the other approaches to onset detection, by filling in a different comparison function. Although we have shown our that comparison structure yields good onset detection even with simple functions, many other techniques, in particular the band-wise approaches cited in section 1, could be used instead. This could prove beneficial particularly if the goal is a standalone onset detection system.

References

- Chang, C.-C. and C.-J. Lin (2001). *LIBSVM: a library for support vector machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Cristianini, N. and J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge University Press.
- de Cheveigne, A. and H. Kawahara (2002). YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America* 111(5), 1917–1930.
- Duxbury, C., J. P. Bello, M. Davies, and M. Sandler (2003). A combined phase and amplitude based approach to onset detection for audio segmentation. In *Proceedings of the European Workshop on Image Analysis for Multimedia Interactive Services*.
- Duxbury, C., M. Sandler, and M. Davies (2002). A hybrid approach to musical onset detection. In *Proceedings of the International Conference on Digital Audio Effects*, pp. 33–38.
- Foster, S., W. A. Schloss, and A. J. Rockmore (1982). Toward an intelligent editor of digital audio: Signal processing methods. *Computer Music Journal* 6(1), 42–51.
- Hainsworth, S. and M. Macleod (2003). Onset detection in musical audio signals. In *Proceedings of the International Computer Music Conference*.
- Klapuri, A. (1999). Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Volume 6, pp. 3089–3092. IEEE.
- Marolt, M., A. Kavcic, and M. Privosnik (2002). Neural networks for note onset detection in piano music. In *Proceedings of the International Computer Music Conference*.
- Scheirer, E. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America* 103(1), 588–601.

²The *F-measure* = $\frac{2PR}{P+R}$ is commonly used in information retrieval to combine precision and recall into a single number.