Auditory modeling is usually based on peripheral physiological phenomena. It is found, however, that this basis is not sufficient in all applications, e.g., in successful speech recognition. Our opinion is that more important therefrom details of periphery is to include higher-level functional processing in the models. This paper describes an experimental system that uses several spectral and temporal representations to create a hierarchical description of speech. The front-end processing is performed by an auditory model, which is based on psychoacoustical principles. Several temporal and spectral representations are extracted from the resulting auditory spectra and are viewed under multiple time resolutions to yield reliable and flexible descriptions of the speech. Based on these spectral and temporal resolutions prominent extrema are located and are classified as objects called events. These objects are organized into event lists according to masking criteria and measures of prominence.

**SYSTEM DESCRIPTION**

The system contains many different levels of processing ranging from auditory modeling of the speech input to symbol and event processing. Figure 1 shows an overview of the current and proposed system. The following sections explain how the system functions.

**Auditory Front End**

The system obtains auditory information from a filter bank which closely matches the human auditory system in terms of sound processing. The model [3] is based on the most important features of peripheral hearing known from the theory of psychoacoustics [4] and simulates the human's frequency selectivity and sensitivity as well as its temporal and masking properties. By the use of this model only relevant auditory information is efficiently removed during the early stages reducing the computation rate in later processing.

The auditory model is implemented as a filter bank and its output is represented by a 48 element spectral vector for each point in time. The vector elements approximate those components of the sound pressure spectrum as it is scaled to loudnesses [4]. Each channel of the filter bank is separated by 0.5 Bark and this provides adequate frequency resolution over the entire 24 Bark auditory spectrum. A spectrum is calculated every 10 ms.

**Multiple Representation Analysis**

The loudness scaled auditory spectra are transformed into several parallel representations which help to identify the different speech features and events. These representations can be separated into two major groups: the frequency domain, and the time domain. These groups are described in the following sections.

**Frequency Domain Processing**

The frequency resolution of the hearing system to broadband signals is at best 1 Bark. For phonetic classification of speech signals different frequency models have shown that this can vary from 1 to 3.5 Bark. We can simulate this effect by bandpass filtering the spectrum in the frequency domain to emphasize the desired resolutions. This bandpass filtered spectrum representation is called the formant spectrum. Adequate resolution has been achieved for this system with both 1 and 2 Bark bandwidth filters. The basis for use of multiple resolutions for a single representation is explained later on. The formant spectrum can be used to identify the existence and locations of formants and formant pairs. Formant lists are created by searching for local maxima and minimum. This representation can be used to detect spectral centers of gravity [5].

**Time Domain Processing**

The other category of representations are based upon information that the front end supplies in the time domain. One such representation is total loudness and is calculated by summing the elements of a loudness spectrum. Total loudness as a function of time reveals the temporal energy structure of the speech while being independent of the individual spectral components.

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple Hierarchy</td>
<td>3.1.2</td>
</tr>
<tr>
<td>Other</td>
<td>3.1.1</td>
</tr>
</tbody>
</table>
Multiple Resolution Analysis

To obtain a more flexible description of each frequency and time-domain representation, all representations are analyzed under several resolutions. This is performed by bandpass filtering a representation with filters having different resolutions. The impulse responses for some of these filters are shown in Figure 4. For the frequency domain representation the loudness spectrum is filtered with 1 and 2 Bark resolutions as was mentioned earlier. In the case the filters are scaled in frequency, in the time-domain representations the filters are scaled in time, and resolutions of the loudness, stationarity, and spectral slope representations are calculated in a similar way. This method is similar to scale-space filtering [6] and is used to generate qualitative descriptions of signals.

![Figure 4. Impulse responses of some of the filters used in Multiple Resolution Analysis.](image)

Each resolution of a representation is defined as a resolution source while the representation along with its resolutions is defined as a resolution group, as indicated in Figure 1. Speech analysis with multiple resolutions facilitates determining event locations and their respective properties with greater ease and accuracy than would be possible with the original representations alone. The curves in Figure 3 show the response of the loudness resolution group to the word lyksil. The multiple/parallel representations and their resolutions allow for a reliable description to be created of the speech. New resolution groups may be added to the system such as pitch, partial classification, and a voiced/unvoiced indicator as is found necessary. The next phase of processing transforms a signal, in this case a resolution source, into a discrete and symbolic representation. The resolution groups are operated upon by event detectors which find local extrema and zero-crossings, depending upon which resolution group is being analyzed, and yield symbols as their output. Symbols are more flexible to manipulate during later stages of processing than signals since partial classification has already taken place. These symbols may contain information regarding their type, time, amplitude and waveform structure. The symbols are ordered chronologically and are placed in a list for later processing.

The resolution group event manager is responsible for analyzing a resolution group and finding the most prominent areas of interest. One measure of prominence is determined by searching for the event with the largest absolute amplitude. It uses as its input the lists of symbols presented to it by the event detector. The resolution group event manager operates on these lists to produce a single list called the resolution group event list that contains the most significant events from a representation.

Since each representation's analysis can be processed independently, parallel-processing of the representations, resolutions, and events is a natural topology for the implementation of such a system. Such a concurrent system could be implemented e.g. using Transputers [10] and is one of our long-range goals.

**CONCLUSION**

Higher-level functional processing must be included in auditory models if the information they supply is to be of greater use. This is because peripheral physiological phenomena often does not offer a sufficient basis for applications such as speech recognition. In this paper we have described an approach to implement the higher-level processing activities into an auditory model. The conversion of speech into a loudness spectrum, the derivation of some representations, and the analysis of these under multiple scale resolutions was explained. Finally, the transformation of signals into discrete frequency/time events was described.

**ACKNOWLEDGEMENTS**

The authors are grateful to the Finnish Academy which has financed this research.

**REFERENCES**


**IMPLEMENTATION**

The preliminary version of the model is currently implemented on a two processor system. The auditory model filterbank is realized on a TMS 320 signal processor and the remainder on an Apple Macintosh. The Macintosh is the host for the TMS and executes NEON which is an object oriented language [7]. NEON is a hybrid language with many of its features derived from Forth and Smalltalk. The next extended version of the program is being currently implemented on a Symbolics 3600 Lap Machine including a small-scale speech recognition system. To efficiently represent and manipulate the different resolutions, representations and symbols, object oriented programming methods are used. Object orientation is a powerful data and knowledge representation principle since knowledge regarding the object is contained within the object itself thus exhibiting object-centered control [8][9]. Objects can communicate with each other by message passing methods. They also belong to classes and inherit properties from other classes. This approach allows for building rule and frame-based systems.