TR-I-0287

Automated Labelling of Prosodic Aspects of English Speech: Final Report

Paul BAGSHAW

1992.11.5

ATR Interpreting Telephony Research Labs.
ATR 自動翻訳電話研究所

Automated Labelling of Prosodic Aspects of English Speech: Final Report

Paul Bagshaw*
ATR Interpreting Telephony Research Laboratories

November 5, 1992

Abstract

Approaches to automatically transcribe the variations in acoustic parameters related to prosodic events in English speech are investigated. The events of interest are the relative prominence of syllables in continuous speech and pitch movements associated with accented syllables. A description is given of a database which contains transcriptions of these prosodic events as perceived by a hand labeller (the "MLP dialogue database"). This database is used in evaluating the performance of the automatic transcription algorithms.

An existing algorithm, the JLH syllable prominence labelling algorithm, is evaluated to give a bench mark against which other approaches are compared. The JLH algorithm is limited by describing syllable prominence as a discrete number of categories and gives no indication of the pitch movements associated with each accented syllable. Furthermore, the approach uses arbitrary thresholds and is rule-based. These limitations and the use of all but a few arbitrary thresholds are overcome without a reduction in performance, by abstracting acoustic parameters related to prosodic events. Phones are grouped into syllables automatically by using a sonorant energy contour. The duration and energy of the phones in these syllables are normalised to compensate for variations attributed to phone type. The fundamental frequency contour is stylised by piece-wise linearisation to remove microprosodic variations. Piece-wise units are related to pitch movements and given relative heights. These abstracted features are mapped to syllable prominence labels using rules similar to those used by the JLH algorithm. Alternative rules are sought by using regression trees to model the hand-transcribed syllable prominence labels given a set of abstracted acoustic features. The tree models do not give a significant increase in the correlation between the automatically transcribed syllable prominences and those transcribed by hand. This is evidence that the rules applied to map the abstracted features to syllable prominence labels are not inappropriate. However, there is an indication that modifications should be made to the rules used to locate accented syllables from the pitch movements.

*Visiting from the Centre of Speech Technology Research, University of Edinburgh

Contents

List of Figures

List of Tables

1 Introduction

The aim of this research is to determine an automatic process of transcribing the variations in acoustic parameters related to prosodic events in English speech. The need for a large corpus of prosodically labelled speech is motivated by the wish to train a speech synthesiser to replicate a variety of prosodic features. The use of prosodically labelled speech is, however, not limited to the training of speech synthesisers. Automatic prosodic transcription also has applications in automated foreign language pronunciation teaching, and can be used to aid parsers in speech recognition systems to disambiguate phonetically similar, but syntactically different utterances.

The prosodic events, whose acoustic variants are to be transcribed, are the relative prominence of syllables in continuous speech and pitch movements associated with accented syllables. In transcribing these prosodic aspects of speech, I aim to only use features which can be reliably extracted automatically from the speech waveform. That is, the amount of information used by the process which relies on human interaction is to be kept to a minimum. For an automatic transcriber of syllable prominence to use knowledge of lexical stress, for example, the word associated with each syllable must be identified. This requires word boundaries to be known, which can only come from an auto-segmentation system if a human interactively stipulates the words of the utterance. Knowledge of lexical stress is therefore not used.

Syllable prominence cannot be reliably described as a simple binary distinction between stressed and unstressed. It is therefore preferable for an automatic prosodic labeller to describe syllable prominence as a relative scalar value. Similarly, pitch movements cannot be categorically classified as a level tone, fall, rise, fall-rise, or rise-fall. Occurrences of "fall-with-a-shallow-rise come fall" and "narrow-rise come level" in pitch are also feasible. Therefore, the automatic transcription of pitch is to describe such movements in terms of a series of relative pitch heights; for example, pitch fall from height $+1.0$ to -0.5 plus a shallow rise to height 0.0. When transcribing syllable prominence and pitch movements by hand, however, it is not possible for a labeller to reliably give scalar descriptors. Even when just seven descriptors are used (level tone, fall, rise, fall-rise, rise-fall, stressed but unaccented or unstressed), one study [19] has shown that two transcribers select the same prosodic label for 72% of syllables.

The method chosen to evaluate an automatic labeller of prosodic events is to compare its transcription with that given by hand labellers. However, there are a number of problems with this approach which must be bore in mind when interpreting the evaluation. Firstly, the automatically determined transcription gives scalar descriptors and the hand transcription gives discrete categories. It is necessary to place thresholds on the scalar descriptors to classify them into one of the discrete categories in order to compare them. The evaluation will therefore depend upon the threshold values. Secondly, the hand labels are not 100% reliable (as indicated by the forementioned study [19]). Finally, it is possible that a human transcriber of the prosodic events does not use acoustic clues alone when judging the perceptual prominence of a syllable. Linguistic knowledge may also be used. For example, when the prominence of a syllable is in question, the labeller may transcribe it as prominent only if it can be lexically stressed. Such knowledge is not be available to the automatic transcription system.

The database of speech used during this research is described in section 2. The contexts of the utterances are described and presented in a series of five sets, of which four are used. The database consists of speech waveforms, automatic phonemic transcriptions, hand labelled prosodic events, and contours of acoustic parameters extracted from the waveforms. The format in which this data exists is described. An evaluation of the JLH syllable prominence labelling algorithm [13] [14] when applied to this data, is presented in section 3. A number of disadvantages of this algorithm are given. Acoustic parameters are mapped to prosodic events by a series of abstractions described in section 4. The aim of this is to overcome the disadvantages of the JLH algorithm. Phones are grouped into syllables automatically and each syllable is given a duration and energy measure. The fundamental frequency contour is stylised to remove microprosodic variations and related to pitch movements. Rules similar to those used in the JLH algorithm are applied to the duration and energy measures, and pitch movements to map them to syllable prominence. The syllable prominences are categorised as accented, stressed or unstressed, so that a comparison can be made with transcriptions made by hand. These rules are investigated in section 5 by using tree-based modelling of the hand transcribed prosodic events.

2 Speech Database of ATR Conference-registration Dialogues with Focus, in English

2.1 Contexts of the utterances

The speech material used during this research is a series of ten telephone dialogues spoken by a female bilingual speaker of Japanese and American English (MLP). A short description of the speaker's background is given in [5]. The context of the speech is that between a receptionist and a potential client for a conference registration. The dialogues are in American English, recorded in a professional, acoustically-damped studio. The "naturalness" of the speech varies from read speech to spontaneous speech over a series of five sets (A to E) of utterances. The first set (A) consists of sentences extracted from the dialogues read with words or phrases "focused"/"emphasised" from a "normal", default reading. The words and phrases with contrastive focus are located at various points in each sentence and are marked in the texts by capitalisation. They are read in order, starting with the default (no focus) reading. The second set (B) consists of the same utterances but read in a randomised order. The third set (C) contains the dialogue sentences, with and without parts focused, when spoken in reply to elicitive questions posed by an interlocutor. The speech for the ten dialogues spoken in a role-playing situation is given in the fourth set (D). The same speaker utters both the roles of the receptionist and the client. The texts of the utterances in this database are shown in appendices A, B, C, and D. Transcription of the fifth set (E) is in process. Sets A, B, C, and D together consist of approximately fifty minutes of speech in a total of 453 utterances. Preparation of sets A, B, C, and D has been part of the work undertaken.

2.2 Data preparation and file formats

2.2.1 The speech waveform

The speech waveforms, sampled at 20kHz and using 16-bit data, have been transferred and reformatted from the CSTR[1] to the ATR file systems. This operation is necessary because of the incompatibility between machine types used at the two sites (SUN machines at CSTR, and DEC machines at ATR). The unreadable "Audlab"[2] header [6] 496-bytes long positioned at the beginning of the CSTR speech files (extension *.vox*) can be stripped off and the necessary byte swapping can be performed, to produce a headerless ATR 20kHz speech file (extension *.sig*) by using the UNIX user command *dd*. A new, readable Audlab header can be attached to the *.sig* files by using the PCB user command *pc2adlb_vox*. This is necessary to use many of the acoustic parameter extraction programs (see the manual pages in appendix E for more details).

2.2.2 Transcriptions - phonemic and prosodic (hand)

The utterances have been segmented into phone units and labelled using a HMM-based automatic segmentation process at CSTR [9]. The phonemic transcriptions for the speech produced by this system are given in ASCII-type files (extension *.trn*). Each line in these files contains at least three fields describing the speech segments. The fields are separated by "white spaces". The first field gives the start time of a segment and the second field gives its stop time. These times are expressed in units of "number of samples" (20,000 samples per second). The third field specifies the phonemic label for a segment using MRPA (Edinburgh University's machine-readable phonemic alphabet). An optional fourth field stores possible diacritics.

Diacritics have been added to the phonemic transcription (in the *.trn* files) to describe prosodic events in the utterances. The prosodic events were transcribed by hand at CSTR using the scheme described in [3]. The hand labels (see table 1) annotate the prominence of syllables defined on a phonological basis and the

[1] Centre for Speech Technology Research, University of Edinburgh
[2] A speech processing package developed at CSTR

Table 1: Symbols for Syllable Prominence and Pitch Movement Labelling

ASCII†	Symbol	Description
u	$\{u\}$	— Completely unstressed
s	$\{s\}$	— Stressed but unaccented
a	$\{a\}$	— Stressed and accented
n	$\{n\}$	— Nuclear accented
| *pipe*	$\{:\}$	— syllable immediately preceding a tone-unit boundary
\	$\{\backslash\}$	— pitch accent is a fall
/	$\{/\}$	— pitch accent is a rise
v	$\{\vee\}$	— accent is a fall-rise
^ *hat*	$\{\wedge\}$	— accent is a rise-fall
l	$\{-\}$	— level tone
<	$\{\leftarrow\}$	— pitch movement is part of the realisation of an accented syllable to the left of this syllable
>	$\{\rightarrow\}$	— pitch movement is part of the realisation of an accented syllable to the right of this syllable
- *minus*	$\{^-\}$	— the range of the pitch movement is unusually wide (increased)
_ *underscore*	$\{\_\}$	— the range of the pitch movement is unusually narrow (decreased)
' *apostrophe*	$\{\triangle\}$	— pitch "peak" or level tone pitch is unusually high
, *comma*	$\{\triangledown\}$	— pitch "peak" or level tone pitch is unusually low
[	$\{\|\ \}$	— initial part of $\{\vee\}$ or $\{\wedge\}$ pitch movement is shallow
]	$\{\ \|\}$	— final part of $\{\vee\}$ or $\{\wedge\}$ pitch movement is shallow

†The ASCII characters listed are the prosodic labels used in machine readable data.

pitch movements associated with the realisation of all accented syllables. Syllables perceived as prominent in the utterance have been labelled as sententially stressed. Syllables have been transcribed as accented (implicitly prominent) at points where there is a pitch discontinuity [8]. The pitch discontinuity perceived to accompany the accented syllable occurs on, before or subsequent to it [11]. The final accented syllable in each tone unit is labelled as nuclear accented (in accordance with the "British School" of intonational phonology [8]). The pitch movement running from one accented syllable to the next or to the end of the tone-unit is transcribed as one of the five categories $\{\backslash, /, \vee, \wedge, -\}$. Pitch range markings are used to describe the extent of the movement; ie. the perceived difference in pitch from its greatest height to its lowest height. If the pitch range is distinctively wider or narrower than expected for a particular contrastive effect, it is marked using the diacritics $\{^-, \_\}$. Diacritics are also applied if the "peak" part of a pitch movement (the initial part of a fall $\{\backslash\}$, the final part of a rise $\{/\}$, and the mid-section of a fall-rise or rise-fall $\{\vee, \wedge\}$) or the pitch of a level tone $\{-\}$ is unusually high $\{\triangle\}$ or low $\{\triangledown\}$ for the particular speaker. In order to describe occurrences of pitch fall-rise and rise-fall with a particularly shallow rise or shallow fall, two further diacritics are employed. These are used to represent, for example, fall-shallow rise as $\{\Psi\}$. All the diacritics in the *.trn* files which describe prosodic events are placed on the first phone label of each syllable.

The diacritics in the *.trn* files are extracted to give two additional label files by using the PCB user command *diac2label*. The first of these files (extension *.str*) describes the phonological-rule-based syllabification and the category of prominence $\{u, s, a, n\}$ for each syllable. A diacritic $\{:\}$ is used in these files to indicate the position of a tone-unit boundary. The second file (extension *.ctr*) describes the pitch movements $\{\backslash, /, \vee, \wedge, -\}$, with diacritics giving additional details. These label files (*.trn*, *.str*, *.ctr*) may be reformated to be compatible with Audlab or the Entropics speech processing software (ESPS and xwaves+) by using the PCB user command *labelformat*.

2.2.3 Acoustic parameters

Sententially stressed syllables are those that are perceived as salient due to a prominence of energy and/or duration and/or pitch [10] [17, chap 4] within an utterance. Therefore, in order to locate such prominences automatically, it is necessary to computed all the correlating acoustic parameters. Those that can be extracted directly from the speech waveform (sampled at 20kHz) without pre-processing are the energy and fundamental frequency. These two parameters are calculated from 20ms frames of

speech data at 5ms intervals so that values are synchronised with the cepstral coefficients and lower three formant frequencies used in the auto-segmentation process. The energy contour is computed by applying the PCB user command *energy* to the raw speech waveform. Each frame is passed through a Blackman-Harris window [12] and the frequency bins of an amplitude spectrum (512-point FFT) corresponding to the range 50Hz–2kHz are accumulated. These energy values are expressed in decibels with respect to the maximum frame energy in the utterance to form an utterance-normalised sonorant energy contour. The fundamental frequency (F0) contour is determined using a slightly enhanced version of the pitch tracker described in [18]. The speech is initially low-pass filtered with a 600Hz -3dB cut-off frequency and more than -85dB rejection above 700Hz. F0 extraction is performed on the low-pass filtered speech by using the PCB user command *srpd*. An evaluation of this PDA [1] has found it to estimate F0 with consistently less than 1% gross pitch errors and less than 16% of speech classified as voiced or unvoiced incorrectly, when compared with laryngeal frequency estimates, Fx. Both the energy contour and the raw F0 contour are processed by a three-frame median filter and five-frame hanning window smoother [20] (PCB user command *smoother*). This removes small perturbations in the energy contour which arise during frames of speech with low fundamental frequency (typically less than two pitch periods per analysis frame), and it eliminates the majority of octave errors and reduces microprosodic perturbations in the F0 contour. The contours are combined into a single Audlab track file (extension *.trk*). The individual tracks can be extracted from this file and the Audlab header removed by using the PCB user command *rm_adlb_hd*.

2.3 Summary

A corpus of 453 utterances with focus has been prepared. The format of the speech waveform (*.sig*) has been described. It has been phonemically transcribed by an automatic segmentation system and prosodic events have been labelled by hand. These transcriptions exist in machine readable formats (*.trn*, *.str*, *.ctr*). A description of the prosodic events annotated by hand has been given. The utterance-normalised sonorant energy contour and fundamental frequency contour (*.trk*) have been extracted from each speech waveform. The techniques used to do this have been described. This corpus is used in our investigation of prosodic events in English speech. Hereinafter, the corpus will be referred to as the "MLP dialogue database".

3 JLH Syllable Prominence Labelling Algorithm: An Evaluation

An implementation of the JLH automatic stress labelling algorithm [13] [14] has been supplied by CSTR. This software has been ported to run on the ATR machine architecture, and used to annotate the prominence of syllables in the MLP dialogue database as either accented $\{n, a\}$, stressed $\{s\}$, or unstressed $\{u\}$. The algorithm uses only the raw speech waveform (*.sig*) and the phonemic transcription (*.trn*). Energy and fundamental frequency extraction are performed internally.

A performance analysis of the JLH rule-based algorithm, over the MLP dialogue database (sets A, B, C and D) was conducted using "Acoustic Phonetics in S" (APS) [23]. The algorithm partitions the utterances and each resulting section is transcribed by one of three stress categories $\{accented, s, u\}$. The labelled sections are compared with the syllable prominence labels (in the *.str* files) for each syllable transcribed by hand. In order to compare these transcriptions, the section which "best" overlaps each syllable is determined. The section covering the greatest proportion of the syllable and which covers at least 50% of the syllable is the section which best overlaps it. The syllable is said to have been automatically transcribed by the label given to the section which best overlaps it. This is how two asynchronous transcriptions are compared. The analysis shows (see table 2) that 59.1% of syllables are correctly annotated as either accented, stressed but unaccented, or unstressed. Unfortunately, 5.9% of the syllables are given no prominence label. The reason for this has not been investigated.

The disadvantages of the JLH algorithm are that it annotates syllable prominence as a discrete number of categories and doesn't give any indication of the type of pitch movement associated with each accented syllable. It does not, therefore meet the aims laid down in section 1. Furthermore, the approach is rule-

Table 2: Confusion Matrix of Syllable Prominence Labels by Hand and by the JLH Algorithm: Sets A, B, C, and D

		JLH Algorithm Label				
		?	*accented*	*s*	*u*	total
Hand Label	*a, n*	49 (0.7%)	1099 (15.1%)	231 (3.2%)	439 (6.0%)	1818 (24.9%)
	s	25 (0.3%)	394 (5.4%)	166 (2.3%)	406 (5.6%)	991 (13.6%)
	u	360 (4.9%)	757 (10.4%)	324 (4.4%)	3049 (41.8%)	4490 (61.5%)
	total	434 (5.9%)	2250 (30.8%)	721 (9.9%)	3894 (53.3%)	7299 (100.0%)

Correct classification rate = 4314/7299 (59.1%)

based with a large number of arbitrary thresholds in addition to parameters which must be determined *a priori* for any given speaker. For example, a maximum of 60% of syllables in any given utterance can be annotated as potentially stressed on the grounds of the duration of their nuclear vowel. Syllables are annotated as potentially stressed only if their utterance normalised sonorant energy exceeds -7dB and they are categorically labelled as unstressed if this energy is below -20dB. Such rules are unfavourable. The JLH algorithm does, however, form a starting point for the investigation and is used as a bench mark against which other approaches may be compared.

4 Abstraction of Acoustic Parameters to Prosodic Events

4.1 Introduction

The aim of the following abstraction is to overcome the disadvantages and limitations of the JLH algorithm, without a reduction in preformance. Phones are grouped into syllables automatically. The duration and energy of the phones in these syllables are normalised to compensate for phone-specific variations and smoothed to iron out dithers in phone boundary placement. A measure of duration and intensity are taken for each syllable. The fundamental frequency contour is stylised to remove micro-prosodic variations and each unit is related to pitch movements. Rules similar to those used in the JLH algorithm are applied to the duration and energy measures, and pitch movements to map them to syllable prominence.

4.2 Automatic syllabification using acoustic parameters

An algorithm to group phones into syllable sized units using the phone boundary and label information given by an automatic segmentation system, and the sonorant energy contour of the utterance is described in [2]. A syllable boundary is located at the phone boundary closest to the local minima in the energy contour between each pair of vocalic or potential syllabic consonantal phones (vowels plus /l, m, n, r[3]/). The nucleus of a syllable is defined as the vowel (or, in the case when there are no vowels, the syllabic consonant) whose associated sonorant energy is greatest. The MLP dialogue database has be syllabified automatically using the algorithm and by hand using a phonologically based syllabification [3]. There is a large correlation between the two resultant syllabic domains (see table 3). "Missing" syllable boundaries are due to the occurrences of vowel/vowel boundaries for which there is no valley in the energy contour between them. Conversely, "extra" syllable boundaries occur when the energy dips within the tenure of the phonologically based syllable at a vowel/vowel boundary or vowel/syllabic consonant boundary.

4.3 Duration and energy measures for syllable prominence

Duration and sonorant energy measures are used in determining the prominence of each syllable in an utterance. The duration and energy variations are mainly attributed to phone type. These parameters

[3] /r/ is included for American English

Table 3: Comparison of Phonologically Based and Acoustically Based Syllabifications

Number of syllables				
from a phonological basis	from the acoustics	Match	Missing	Extra
7299 (100.0%)	7011 (96.1%)	6980 (95.6%)	-319 (4.4%)	+31 (0.4%)

are therefore Z-score normalised with respect to the phone type in order to compensate for segmental variations [4]. The mean duration and energy, and their population standard deviations are determined for each phone type from a training database of 200 phonemically balanced utterances[4] spoken by MLP (see PCB user command *prosody_stats*). The Z-score normalisation of a phone's duration or sonorant energy simply involves subtracting the mean value and dividing by the population standard deviation for that phone type. Sections of speech labelled as silence are coded with a normalised duration and normalised energy of -9999.0 ("break number"). The resultant phone-level, normalised duration and energy contours are non-linearly smoothed using a three-phone median filter and a three-phone hanning window [20]. Special treatment of the "break number" is analogous to that in the PCB user command *smoother* used for frame-level contours. The maximum, smoothed, normalised duration and energy of the phones in each syllable are used as a measures in determining its prominence.

4.4 Abstraction of fundamental frequency to pitch movements

4.4.1 Overview

Fundamental frequency is related to pitch movements by a two-stage abstraction. The first stage of abstraction is a linear piece-wise stylisation of an F0 contour (in the *.trk* file) using the robust least median of squared residuals regression analysis (LMedR) [21]. This stylisation aims to eliminate microprosodic variations in the contour. Each piece-wise unit forms a possible pitch movement or part of a movement. The second stage of abstraction relates these units to pitch movements. Only those units which occur during a syllable nucleus, in part or in whole, are selected. The pitch units may therefore extend beyond the nucleus but only those crossing the nucleus are reliable. This approach therefore compromises between using information about the movement of F0 through vowels alone (which may be limiting for short nuclei), and using the F0 contour of an entire syllable (where F0 discontinuity errors may occur). Each movement is described by the pitch height at its beginning and its end relative to an utterance dependent datum.

4.4.2 The formation of a piece-wise F0 contour

The algorithm used to perform the stylisation of an F0 contour is based on the technique described in [22] and incorporates LMedR analysis. The F0 values describing the contour (excluding values which equal zero to represent unvoiced speech) are converted to the semitone scale using the relationship $F0_{semitone} = 12\log_2(F0_{hertz}/55)$. Significant turning-points in the F0 contour are located, these points are modified to prevent contour discontinuities other than at the boundaries between unvoiced and voiced speech, and a new contour is generated by interpolating between them.

The following process is used to identify the turning-points. Starting with the first voiced frame, LMedR analysis is applied to a window of w frames corresponding to voiced speech, where w is initially set to 5. The final frame in this window is taken to be a turning-point candidate. The F0 value of the subsequent frame is predicted using the coefficients of the LMedR analysis. If the absolute difference between the actual and predicted F0 values is less than or equal to some level of permitted variation in F0 (1 semitone), then the candidate is not a turning-point, the window length w is incremented to include the next voiced frame, and the above process is repeated. The repetition of this process terminates when the turning-point candidate is the final voiced frame in the F0 contour. Otherwise, when the absolute difference is greater than the permitted F0 variation, either this subsequent F0 value constitutes some

[4] The texts of these utterances are the same as those used in the ATR/CSTR Speech Database Project

type of irregularity in the F0 contour or the candidate could be a true turning-point. To determine which is the case, the F0 value of the next voiced frame is also predicted. If the absolute difference between the predicted and actual values is once again greater than the permitted F0 variation, and this situation arises for all following frames up to either the final voiced frame in the contour or such that the duration of this discontinuity is greater than some minimum permitted level (100ms), which ever occurs first, then the candidate is said to be a true turning-point. Otherwise, the length of the window w is increased to include the first frame for which the absolute difference in actual and predicted F0 values was less then or equal to the permitted variation, but not those for which it was greater, and the LMedR analysis process is repeated. If the candidate was found to be a turning-point and if it corresponds to a voiced frame immediately preceding a frame of unvoiced speech, then the first frame of the next voiced region is also designated as a turning point. This process is then repeated with the length of the window w reset to 5 and the first frame of the window is set to the frame of the most recent turning-point found. The first and final voiced frames of the non-stylised contour are also assigned as turning-points.

In order to ensure that discontinuities in the stylised F0 contour only occur at unvoiced sections of speech, the fundamental frequency at each turning-point of the new contour is determined in a way which depends upon the voicing state of the frames adjacent to it. For any given turning-point (tp) at frame f_{tp} with original fundamental frequency $F0_{tp}$, the LMedR coefficients s_{tp} (slope) and i_{tp} (intercept) of the windowed points preceding the turning-point are known. The modified fundamental frequency $F0'_{tp}$ is given as,

$$F0'_{tp} = \begin{cases} 0.5(s_{tp}.f_{tp} + i_{tp} + s_{tp+1}.f_{tp} + i_{tp+1}) & \text{if frames } f_{tp}-1 \text{ \& } f_{tp}+1 \text{ voiced} \\ s_{tp+1}.f_{tp} + i_{tp+1} & \text{if frame } f_{tp}-1 \text{ unvoiced \& frame } f_{tp}+1 \text{ voiced} \\ s_{tp}.f_{tp} + i_{tp} & \text{if frame } f_{tp}-1 \text{ voiced \& frame } f_{tp}+1 \text{ unvoiced} \\ F0_{tp} & \text{if frames } f_{tp}-1 \text{ \& } f_{tp}+1 \text{ unvoiced} \end{cases} \quad (1)$$

The new stylised contour is then created by linear interpolation of F0 between each turning-point (f_{tp}, $F0'_{tp}$) and by reseting each frame that is unvoiced in the non-stylised contour to an unvoiced state in the new one. The resultant data is then coverted back to a Hertz scale.

The F0 contours produced for the MLP dialogue database have been stylised using this method (PCB user command *piecewise*).

Cepstral-based resynthesis of speech using the smoothed fundamental frequency contour (in the *.trk* file) extracted from the raw speech waveform, demonstrates the reduction in speech quality due to the resynthesis technique [16] [15] (PCB user commands *cep_fft* and *cep_syn*). This resynthesised speech has been compared, by ear, with speech resynthesised using a piece-wise linearised F0 contour ("piece-wise-F0-speech") and using the residual F0 (difference between the original and linearised F0s) offset by the mean fundamental frequency of the utterance ("residual-F0-speech"). This comparison is aimed to determine if the linearisation has succeeded in isolating the intonational variations of F0 from the segmentally imposed variations. The residual-F0-speech sounds almost intonationally flat. However, since there is an interaction between the segmentals of speech and prosody, variations in pitch are perceived. The durational and energy variants are also present in the residual-F0-speech, giving considerable prosodic detail to it. The durational and energy variants also exist in the piece-wise-F0-speech, and thus mask the elimination of intonational variations of F0 due to the piece-wise linearisation. It is therefore difficult to say whether or not the piece-wise linearisation has succeeded in isolating the intonational variations of F0 form the segmentally imposed variations. However, of the 453 utterances in the MLP dialogue database, 405 (89.4%) were heard, by my, to contain no perceptual difference in prosodic content. Ideally a perceptual experiment should be conducted to evaluate this F0 stylisation.

4.4.3 From piece-wise units to pitch movements

The piece-wise F0 contour contain, for some utterances, some units which are erroneous, ie. do not correspond to part of pitch movements. Only those piece-wise units which, at some time, run through any part of a syllable nuclear phone (where F0 estimation is expected to be reliable) are treated as being part of a pitch movement. Moreover, the absolute F0 range of a piece-wise unit is not of interest as it will vary from speaker to speaker, but its extent relative to other units in the utterance is. The relative extent of each piece-wise unit is calculated by first locating a regression line which best fits the contour

turning points using LMedR analysis. A by-product of the LMedR is the standard deviation, σ_{LMedR} of the points from the resultant linear model. The absolute F0 at each turning point is then converted by subtracting its modelled value and dividing by the standard deviation, σ_{LMedR}. This effectively compensates for any long term declinative tendency that may be exhibited in the fundamental frequency contour, and expresses the F0 values relative to an utterance dependent datum.

Once the relative extent of each piece-wise unit has been established, they are combined to form pitch movement descriptors. The pitch movements facilitated are level, fall, rise, fall-rise and rise-fall $\{-, \backslash, /, \vee, \wedge\}$, as given by the "British School" of intonational phonology [8]. Each piece-wise unit crossing any part of a syllable nuclear phone is classified as either level, fall, or rise. Let $F0_{start}$ be the relative F0 height at the start of the piece-wise unit and that at the end of the unit be $F0_{end}$. The piece-wise unit is classified on the following basis,

$$\text{pitch movement} = \begin{cases} \backslash & \text{if } F0_{start} - F0_{end} > 0.75\sigma_{LMedR} \\ / & \text{if } F0_{start} - F0_{end} < -0.75\sigma_{LMedR} \\ - & \text{otherwise} \end{cases} \quad (2)$$

When more than one piece-wise unit crosses any particular nucleus, they are combined by initially taking all adjacent units with the same pitch movement classification and joining them into one. A join is made by setting $F0_{start}$ to that of the first unit, $F0_{end}$ to that of the second unit, and reclassifying using equation 2. In the MLP dialogue database of 453 utterances, consisting of 7299 syllables, there were only 4 syllables for which more than two units remained after this process. These all contained some error which originated in the F0 estimation. If there are two remaining units (their classifications must differ), and if either is classified as level $\{-\}$, then they too are joined in the same way. Otherwise, one is a fall $\{\backslash\}$ and the other is a rise $\{/\}$. These are combined to give a single movement classified as either a fall-rise $\{\vee\}$ or rise-fall $\{\wedge\}$ depending on their order, and the relative height at their mid-point is kept. Thus, for the fall-rise and rise-fall classifications, the relative height of both the onset and coda of the movement are known.

4.5 Example of abstraction

An example of these processes is shown in figure 1. Part (a) shows the speech waveform and its corresponding automatic segmentation using MRPA labels. Part (b) gives the utterance normalised sonorant energy contour and transcription-aligned syllable boundaries with a MRPA label indicating the phone forming the nucleus. There is one example of a "missing" syllable boundary in the automatic syllabification of "international". Syllabification using phonological rules segments this word into five syllables. However, only four "pockets" of sonorant energy are formed when it is spoken. The automatic syllabification therefore segments this word into four syllables. The Z-score normalised, smoothed duration and energy measures for each syllable are given in part (c). The fundamental frequency contour and its piece-wise stylised form are shown in part (d). At some points, there is a large difference (up to 50Hz) between the original F0 contour and the piece-wise F0 contour. Large pitch variations therefore exist in the residual-F0-speech. However, such variations in pitch do not mean that the prosodic variants of F0 are not represented in the piece-wise F0 contour. In this example, the piece-wise F0 contour can be clearly seen to contain three phrase units corresponding to "The conference will take place from", "August 22nd to 25th at", and "the Kyoto international conference centre."

4.6 Evaluation

4.6.1 Mapping acoustic features to syllable prominence

A duration measure and an energy measure are known for each syllable in an utterance. A description of the pitch movement associated with each syllable is also known. These features are used to categorise each syllable as accented, stressed but unaccented or unstressed so that the resultant transcription may be compared with those given by hand. Three factors are used in categorising a syllable's prominence, named "*duration factor*", "*energy factor*" and "*pitch factor*". Each factor has one of three possible values, "*Max*", "*s*", or "*u*", defined as follows.

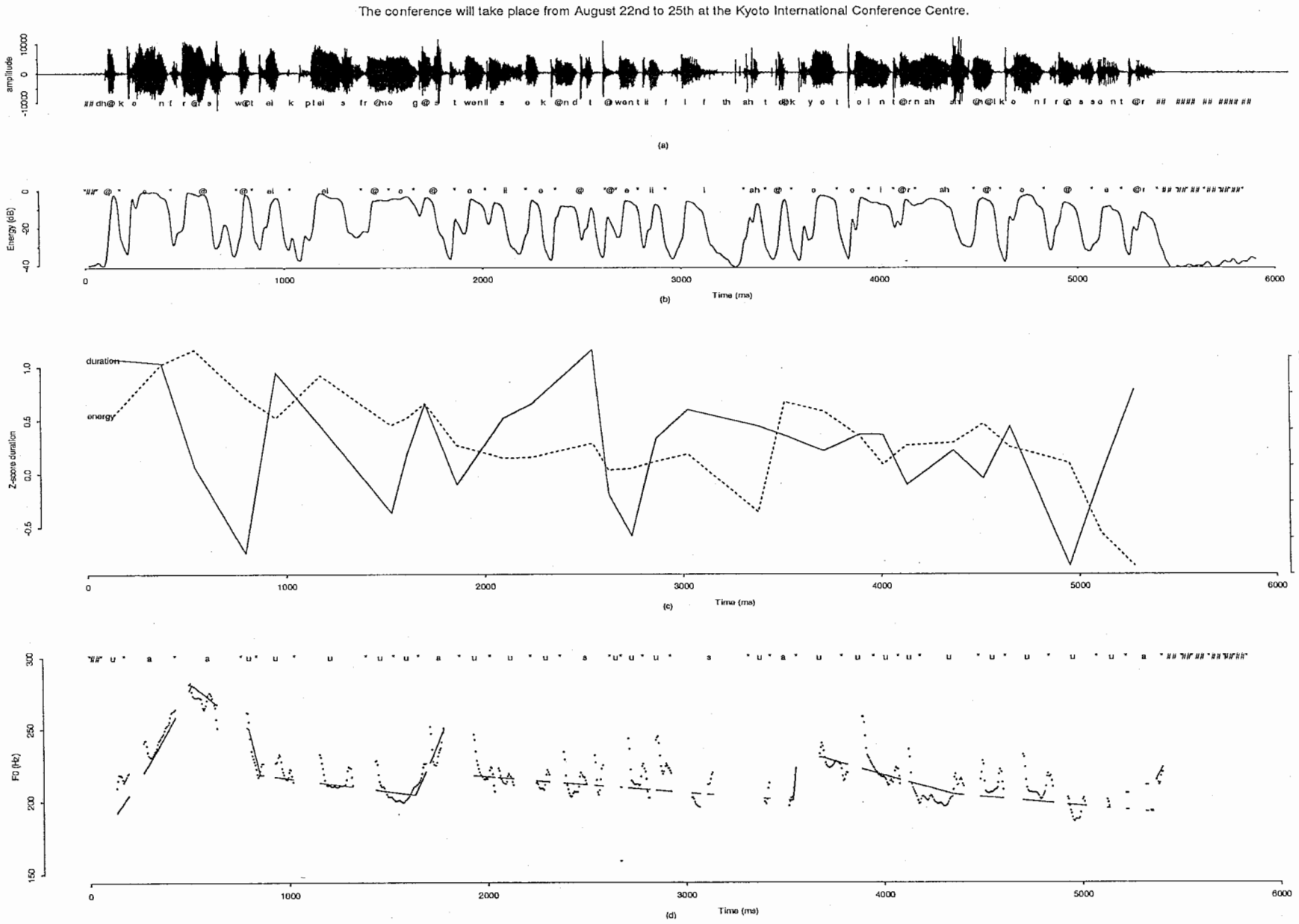


Figure 1: Example of the abstraction of acoustic features related to prosodic events

Table 4: Confusion Matrix of Prosodic Transcription by Hand and by Automation

		Automatic Label			
		accented	*s*	*u*	total
Hand Label	*a*	660 (9.0%)	216 (3.0%)	942 (12.9%)	1818 (24.9%)
	s	165 (2.3%)	91 (1.2%)	735 (10.1%)	991 (13.6%)
	u	558 (7.6%)	296 (4.1%)	3636 (49.8%)	4490 (61.5%)
	total	1383 (18.9%)	603 (8.3%)	5313 (72.8%)	7299 (100.0%)

Correct classification rate = 4387/7299 (60.1%)

- *"Duration Factor"*
  If the duration measure for a syllable is the maximum for the utterance, the *duration factor* is set to *"Max"*. It is set to *"s"* if the duration measure for the syllable is greater than that of both its nearest neighbours (end-points being inherently lower), and it is greater then 0.0 standard deviations from the mean value. (The value 0.0 is an arbitrary threshold). Otherwise the *duration factor* is set to *"u"*.

- *"Energy Factor"*
  If the energy measure for a syllable is the maximum for the utterance, the *energy factor* is set to *"Max"*. It is set to *"s"* if the energy measure for the syllable is greater than that of both its nearest neighbours (end-points being inherently lower), and it is greater then 0.0 standard deviations from the mean value. (The value 0.0 is an arbitrary threshold). Otherwise the *energy factor* is set to *"u"*.

- *"Pitch Factor"*
  If the pitch height is the maximum for the utterance, the *pitch factor* is set to *"Max"*. It is set to *"s"* if the syllable is pitch salient according to the JLH decision filter which is three pitch movements wide (described in [13]). Otherwise the *pitch factor* is set to *"u"*.

A syllable is categorised as prominent if two out of three of the above factors are equal to *"s"* or if any one of them is equal to *"Max"*. A syllable is categorised as pitch salient if the *pitch factor* equals *"s"* or *"Max"*. Non-prominent syllables are labelled as $\{u\}$, prominent syllables which are pitch salient are labelled as $\{accented\}$ and the remaining prominent syllables are labelled as $\{s\}$. This categorisation of each syllable is similar to the method used in the JLH algorithm. An example of such prominence detection is illustrated at the top of figure 1(d).

4.6.2 Application to the MLP dialogue database

The MLP dialogue database has been automatically prosodically marked in this way (using the PCB user command *auto_prosody*) and is compared with those transcribed by hand (see table 4). The transcriptions are equivalent for 60.1% of the syllables. The overall correct classification rate is comparable with that of the JLH algorithm. The advantages at this point, however, are that few arbitrary thresholds are involved and relative pitch prominence values can be obtained.

Of the unstressed $\{u\}$ hand labels marked as either accented or stressed automatically, 194 were syllables with a schwa nucleus. This indicates that the hand transcriber may be marking syllables as sententially stressed only if they can be lexically stressed. The correlation would increase to 62.8% by adding the rule, "classify all schwa nuclear syllables as unstressed". There is also a noticeably large number of syllables labelled by hand as accented or stressed that are marked automatically as unstressed, indicating that the hand labeller may be using acoustic parameters other than those described previously in this paper. For example, a syllable whose nucleus is "fully articulated" is often hand labelled as stressed. Such measures are currently unavailable to the automatic prosodic transcription algorithm. Future research should include further analysis of these errors.

4.7 Summary

Phones in the MLP dialogue database have been grouped into syllables using an algorithm based on acoustic features. The resultant syllabification correlates closely with a phonologically based syllabification. The piece-wise stylisation of a fundamental frequency contour to eliminate micro-prosodic variations in F0 has been further abstracted to form pitch movements for each syllable. The extend of these movements are known relative to other pitch movements in the utterance. The prominence of each syllable has been determined from these parameters and compared with a hand-labelled prosodic transcription. The correlation between labels (60.1%) is lower than one would hope, and possible reasons for this have been discussed. Further analysis of these results is required. The rules used in mapping the acoustic features to syllable prominence are a contributing factor to these errors. The following section investigates the rules used.

5 Tree-based Modelling of Prosodic Events.

5.1 Motivation for using a regression tree

A tree-based statistical model has been used to investigate the relationship between a set of acoustic features and the hand transcribed syllable prominence labels in the MLP dialogue database. There are several reasons for choosing a regression tree for this study.

- An implementation of a classification and regression tree (CART) algorithm is available in the widely used statistical analysis package, "S" (version 3.0)[5] [7].
- A regression tree can be used to model a single-response variable given a set of predictor variables. In this case, the response variable is a multi-level factor – the category of syllable prominence ($\{accented, s, u\}$).
- A number of different acoustic features are selected as predictor variables in training a regression tree. To investigate the effect of introducing an additional feature on the classification of the response variable, the model must be retrained and its performance evaluated. A regression tree can easily be retrained to classify the response variable given a new set of predictor variables.
- The predictor variables applied to a regression tree can contain a mixture of numeric variables and factors.
- The CART algorithm gives a disjunctive set of decision rules, represented in the form of a binary decision tree. These rules are applied to the predictor variables to map to a category of syllable prominence. These rules can be interpreted and compared with those used in the JLH algorithm (section 3).

We wish to investigate the relationship between a set of acoustic features and syllable prominence. A speech database containing reliable syllable prominence labels is needed for a regression tree to model this relationship. The syllable prominence labels in the MLP dialogue database are used in training the tree-based model. However, it has already be mentioned (section 1) that such labels transcribed by hand are not 100% reliable. The regression tree's ability to model syllable prominence will therefore be limited.

The regression trees are used to model accented (nuclear or otherwise), stressed but unaccented, and unstressed syllables, given a set of predictor variables. The various categories of accented syllable ($\{\backslash, /, \vee, \wedge, -\}$) have been merged into one category ($\{accented\}$) in order to simplify the model and its subsequent interpretation.

[5]The CART algorithm implementation has been assumed to function correctly. However, "S" has failed to operate correctly given some tree growing tasks. The reliability of this software is therefore in question.

Table 5: Confusion Matrix of Syllable Prominence Labels by Hand and by Tree modelling: Sets A, B, C, and D – 3-level factors as predictor variables

		Tree Predicted Label			
		a,n	*s*	*u*	total
Hand Label	*a,n*	349 (4.8%)	0 (0.0%)	1469 (20.1%)	1818 (24.9%)
	s	70 (1.0%)	0 (0.0%)	921 (12.6%)	991 (13.6%)
	u	203 (2.8%)	0 (0.0%)	4287 (58.7%)	4490 (61.5%)
	total	622 (8.5%)	0 (0.0%)	6677 (91.5%)	7299 (100.0%)

Correct classification rate = 4636/7299 (63.5%)

5.2 Application of a regression tree to model syllable prominence

Two recession trees have been trained to map a set of acoustic features to syllable prominence labels. The first of these trees gives rules to be compared with the "two-out-of-three or maximum" rule used in section 4.6.1. The second tree attempts to model syllable prominence given less abstract acoustic features.

5.2.1 Duration, energy and pitch 3-level factors as predictor variables

The tree is trained to model the syllable prominence labels in the MLP dialogue database (the *.str* files) using the three factors, "*duration factor*", "*energy factor*" and "*pitch factor*" (defined in section 4.6.1), as predictor variables. The tree (see figure 2) contains 24 nodes, has a residual mean deviation of 1.742 and correctly classifies 63.5% of the syllable prominences. The confusion matrix in table 5 shows the classification in more detail. In summary, the tree classifies a syllable as accented under the following conditions:

$$\begin{array}{ll} a. & pitch\,factor == \text{“}Max\text{”}\ AND\ duration\,factor != \text{“}u\text{”}\ AND\ energy\,factor != \text{“}u\text{”} \\ b. & pitch\,factor == \text{“}s\text{”}\ AND\ (duration\,factor == \text{“}Max\text{”}\ OR\ energy\,factor != \text{“}u\text{”}) \\ c. & pitch\,factor == \text{“}u\text{”}\ AND\ duration\,factor == \text{“}Max\text{”}\ AND\ energy\,factor != \text{“}u\text{”} \end{array} \quad (3)$$

Otherwise, the syllable is classified as unstressed. No syllables are classified as stressed but unaccented, and there are noticeably few accented syllables correctly modelled (349) compared with those in table 4 (660). The last of the above three conditions (equation 3c) indicates that accented syllables might exist for which the *pitch factor* equals "*u*". The JLH decision filter which is three pitch movements wide (used in forming the *pitch factor*) should therefore be modified. Note, however, that some (unknown number) of the syllables transcribed as accented by the tree using this rule are labelled as unaccented by hand (ie. some of the 70 + 203 errors). Further research should be conducted to investigate the relationship between the piece-wise units in F0 and the location of accented syllables.

5.2.2 Acoustic features as predictor variables

A number of acoustic features extracted from the speech waveform are used as the predictor variables. Twelve features are used to describe duration, energy and fundamental frequency over a three-syllable window. Two features are used to capture the inflections of F0 in each syllable, namely the relative F0 heights at the beginning of the syllable ("*left.f0*") and at the end of the syllable ("*right.f0*"). For each syllable, there are four acoustic features extracted – smoothed, phone-normalised duration of the syllable nucleus ("*duration*"), the maximum, smoothed, phone-normalised energy in the syllable ("*energy*"), "*left.f0*", and "*right.f0*". The abstraction of the acoustic parameters, duration, energy and fundamental frequency is described in section 4. In automatically establishing the prominence of any syllable in an utterance, these four features for the current, previous ("*l.duration*", "*l.energy*",

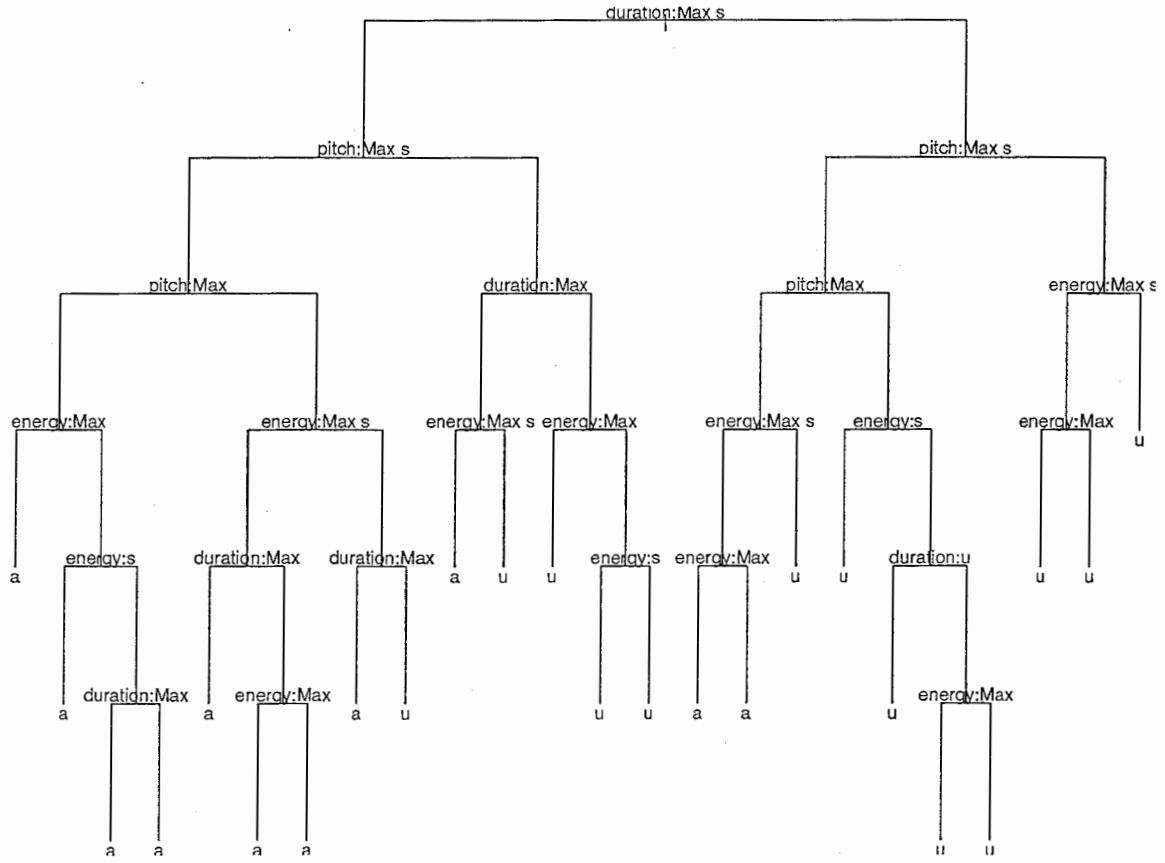


Figure 2: Tree modelling syllable prominence on 3-level factors

Table 6: Confusion Matrix of Syllable Prominence Labels by Hand and by Tree modelling: Sets A, B, C, and D – fully grown tree

		Tree Predicted Label			
		a,n	*s*	*u*	total
Hand Label	*a,n*	798 (10.9%)	45 (0.6%)	975 (13.4%)	1818 (24.9%)
	s	155 (2.1%)	105 (1.4%)	731 (10.0%)	991 (13.6%)
	u	304 (4.2%)	55 (0.8%)	4131 (56.6%)	4490 (61.5%)
	total	1257 (17.2%)	205 (2.8%)	5837 (80.0%)	7299 (100.0%)

Correct classification rate = 5034/7299 (69.0%)

Table 7: Confusion Matrix of Syllable Prominence Labels by Hand and by Tree modelling: Sets A, B, C, and D – pruned tree

		Tree Predicted Label			
		a,n	*s*	*u*	total
Hand Label	*a,n*	463 (6.3%)	0 (0.0%)	1355 (18.6%)	1818 (24.9%)
	s	114 (1.6%)	0 (0.0%)	877 (12.0%)	991 (13.6%)
	u	226 (3.1%)	0 (0.0%)	4264 (58.4%)	4490 (61.5%)
	total	803 (11.0%)	0 (0.0%)	6496 (72.6%)	7299 (100.0%)

Correct classification rate = 4727/7299 (64.8%)

"$l.left.f0$", and "$l.right.f0$") and next ("$r.duration$", "$r.energy$", "$r.left.f0$", and "$r.right.f0$") syllables are used, giving a total of twelve features per syllable.

The tree modelling the syllable prominences in the MLP dialogue database, given these features, correctly classifies 69.0% of the prominences. The confusion matrix in table 6 shows the classification in more detail. The induced decision tree, given the entire database, almost certainly "overfits" the learning set. Such a tree is unnecessarily complex (in this case, containing 193 nodes), making it both less interpretable and more error prone than a tree of "the right size" since it is too specific to the training data. In order to determine "the right size" of the tree, cross validation and tree pruning [7] have been employed. The syllables in the database are put into one of ten sets by random. A tree model is trained on nine of these sets. The model is then pruned using a range of "k factors" and each resultant pruned tree is used to predict the syllable prominences of the remaining set. This is repeated in a cyclic fashion for each of the ten sets. The error in predicting the excluded set is accumulated for each k factor. A measure of the prediction error is the "residual mean deviance". The k factor corresponding to the minimum residual mean deviance gives the pruned tree of "the right size". Figure 3 gives the result of this process when applied to the MLP dialogue database. The minimum deviance corresponds to a k factor of 26.23. When a tree model, trained using the entire database, is pruned using this factor, the 14-node tree shown in figure 4 is formed. The confusion matrix (table 7) indicates the number of occurrences that each hand-transcribed label is classified as accented $\{a, n\}$, stressed $\{s\}$ or unstressed $\{u\}$ using this pruned tree. 64.8% of the syllables are correctly classified. In summary, the tree classifies a syllable as accented under the following conditions:

$$\begin{aligned} &a.\ duration > 1.17\, AND\, r.energy < -3.0\, AND\, right.f0 < 0.045 \\ &b.\ duration < 0.34\, AND\, right.f0 > -0.035\, AND\, left.f0 < -0.115\, AND\, energy > 0.50 \\ &c.\ duration > 0.34\, AND\, right.f0 > 0.045\, AND\, (l.right.f0 < -0.125\, OR\, energy > 0.68) \end{aligned} \quad (4)$$

Otherwise, the syllable is classified as unstressed. No syllables are classified as stressed but unaccented. The correct classification rate (64.8%) obtained by applying the rules in equation 4 is only slightly better than that using the rules in equation 3. This suggests that the categorisation of acoustic features to syllable prominences described in section 4.6.1 is not inappropriate.

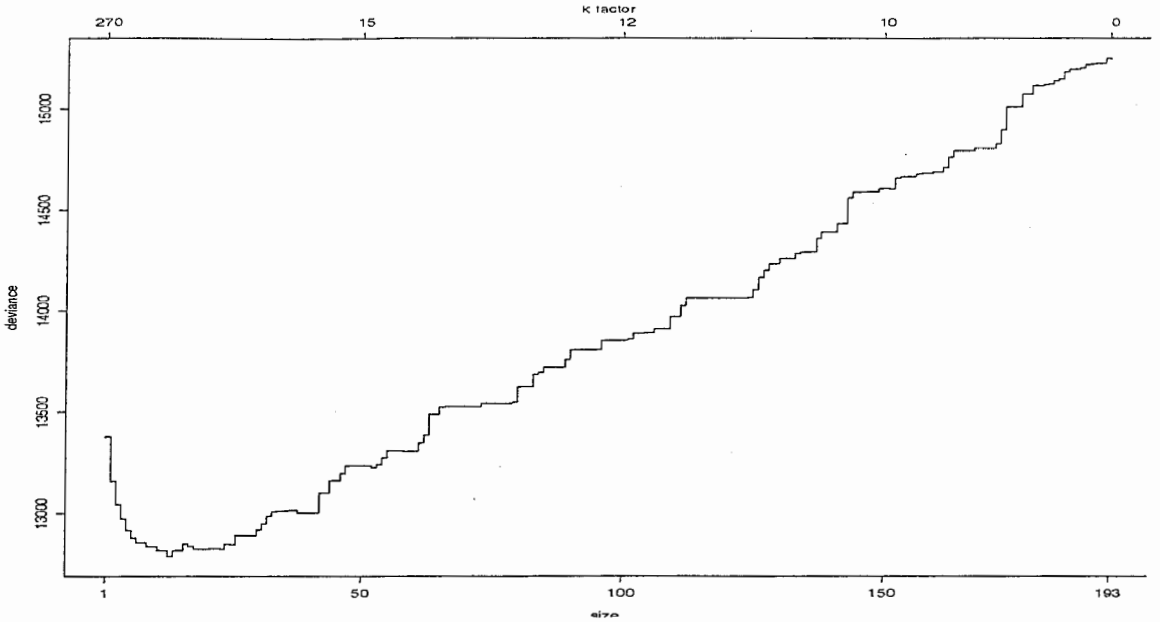


Figure 3: Tree sequence obtain by cross validation

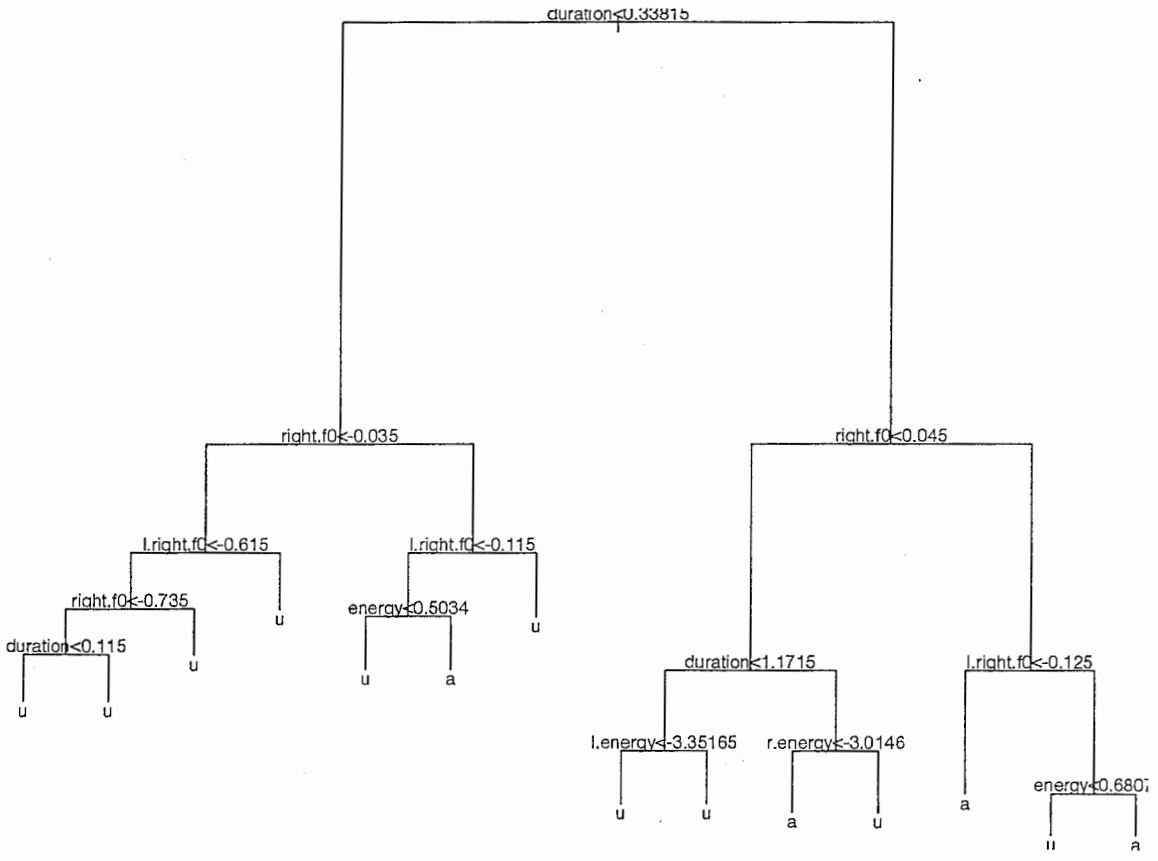


Figure 4: Pruned tree modelling syllable prominence on acoustic features

5.3 Summary

Syllable prominences labelled by hand in the MLP dialogue database have been modelled using the CART algorithm. The prominence labels have been modelled on the three factors, "*duration factor*", "*energy factor*", and "*pitch factor*" (defined in section 4.6.1). The rules from the resultant tree indicate that modifications should be made to the JLH decision filter used to locate accented syllables from the pitch movements. The prominence labels have also been modelled on a set of abstracted acoustic features. Neither of these tree models give a significant increase in the correlation between automatically and hand transcribed syllable prominence labels. This is evidence that the rules described in section 4.6.1 are not inappropriate.

6 Conclusion

A series of abstractions are applied to the acoustic parameters of a speech waveform in order to map them to prosodic events. The abstraction involves compensating for variations in the parameters attributed to segmental characteristics. The abstracted acoustic features are mapped to syllable prominence labels for comparison with those transcribed by hand. There is a correlation of 60.1% between automatically and hand labelled syllable prominences. The rules used to map these features to the syllable prominence labels have been investigated using tree-based models. The trees have shown little improvement (63.5% and 64.8% correlation) over the rules used, suggesting that they are not inappropriate. The abstraction of the acoustic parameters also describes syllable prominence as a scalar and gives pitch movements with pitch heights relative to an utterance dependent datum. The abstraction uses only few arbitrary thresholds. This approach overcomes many of the problems in the JLH algorithm, without a reduction in performance. The JLH algorithm gave a correlation between automatically and hand transcribed syllable prominence labels of 59.1%.

Acknowledgements

I would like to express my thanks to Dr. Kurematsu for his kind support of my stay at ATR and Prof. Mervyn Jack for facilitating my visit from CSTR. Further thanks to Nick Campbell and Yoshinori Sagisaka for their supervision of the work undertaken during the six months of the research. Thanks also to Briony Williams, Jacqueline Vaissière, Jim Hieronymus, Peter Meer, Keith Edwards, Sally Bates, Alex Monaghan, and Bob Ladd, all of whom have, at some stage, contributed valuable assistance and suggestions. The work has been kindly supported by ATR Interpreting Telephony Research Laboratories, Kyoto, Japan. Many thanks also go to the people of ATR for making the six month period of my visit an enjoyable one.

References

[1] P.C. Bagshaw. On the determination of fundamental frequency in speech signals for the automatic analysis of prosody. Technical report, Centre for Speech Technology Research, University of Edinburgh, U.K., 1991.

[2] P.C. Bagshaw. An investigation of acoustic events related to sentential stress and pitch accents, in English. In *Proc. 4th. Australian International Conference on Speech Science and Technology*, Brisbane, Australia, 1992. (in press).

[3] P.C. Bagshaw and B.J. Williams. Criteria for labelling prosodic aspects of English speech. In *Proc. International Conference on Spoken Language Processing*, volume 2, pages 859–862, Banff, Canada, 1992.

[4] W.N. Campbell. Evidence for a syllable-based model of speech timing. In *Proc. International Conference on Spoken Language Processing*, volume 1, pages 9–12, Kobe, Japan, 1990.

[5] W.N. Campbell. Prosodic encoding of English speech. In *Proc. International Conference on Spoken Language Processing*, volume 1, pages 663–666, Banff, Canada, 1992.

[6] Centre for Speech Technology Research, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, Scotland. *Audlab User Manual*, 3.1 edition.

[7] L.A. Clark and D. Pregibon. Tree-based models. In J.M. Chambers and T.J. Hastie, editors, *Statistical Models in S*, chapter 9, pages 377–419. Wadsworth & Brooks, Pacific Grove, California, 1992.

[8] D. Crystal. *Prosodic Systems and Intonation in English.* Cambridge University Press, Cambridge, U.K., 1969.

[9] K. Edwards, M.S. Schmidt, and M.A. Jack. Evalutation of an HMM-based automatic speech segmentation system applied to ATR speech data. Technical report, Centre for Speech Technology Research, University of Edinburgh, U.K., 1992. (prepared for ATR).

[10] D.B. Fry. Duration and intensity as physics correlates of linguistic stress. *Journal of the Acoustical Society of America*, 27(4):765–768, 1955.

[11] E. Gårding and Gerstman. The effect of changes in the location of an intonation peak on sentence stress. *Studia Linguistica*, 14:57–59, 1960.

[12] F.J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proc. IEEE*, 66(1):51–83, 1978.

[13] J.L. Hieronymus. Automatic sentential vowel stress labelling. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH-89)*, volume 1, pages 226–229, Paris, 1989.

[14] J.L. Hieronymus and B.J. Williams. An investigation of the relation between perceived pitch accent and automatically-located accent in british english. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH-91)*, volume 3, pages 1157–1160, Genova, Italy, 1991.

[15] S. Imai. Log magnitude approximation (LMA) filter. *IEICE Trans. Acoustical Society of Japan*, J63–A(12):886–893, 1980. (in Japanese).

[16] S. Imai and Y. Abe. Spectral envelope extraction by improved cepstral method. *IEICE Trans. Acoustical Society of Japan*, J62–A(4):217–223, 1979. (in Japanese).

[17] I. Lehiste. *Suprasegmentals.* The Massachusetts Institute of Technology Press, Cambridge, Massachusetts, 1970.

[18] Y. Medan, E. Yair, and D. Chazan. Super resolution pitch determination of speech signals. *IEEE Trans. Signal Processing*, ASSP–39(1):40–48, 1991.

[19] B. Pickering, B. Williams, and G. Knowles. Analysis of transcriber differences in the SEC. In P. Alderson and G. Knowles, editors, *Working with speech*, chapter 4. Longman, London, (in press).

[20] L.R. Rabiner, M.R. Sambur, and C.E. Schmidt. Applications of non-linear smoothing algorithms to speech processing. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP–23(6):552–557, 1975.

[21] P. Rose. Considerations in the normalisation of the fundamental frequency of linguistic tone. *Speech Communication*, 6(4):343–352, 1987.

[22] M.T.M. Scheffers. Automatic stylization of F0-contours. In W.A. Ainsworth and J.N. Holmes, editors, *Proc. of 7th. FASE Symposium*, volume 3, pages 981–987, Edinburgh, 22–26 August 1988.

[23] G.S. Watson. APS: an environment for acoustic phonetic research. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH-89)*, volume 2, pages 300–303, Paris, 1989.

A Text of Database Utterances - set A

001 The proceedings and the reception are included in the registration fee.
002 The proceedings AND THE RECEPTION are included in the registration fee.
003 The proceedings and the reception are included in the REGISTRATION fee.
004 There's no discount this time.
005 There's NO discount this time.
006 There's no discount THIS time.
007 Payment should be made by bank-transfer.
008 Payment should be made by BANK-TRANSFER.
009 Payment SHOULD be made by bank-transfer.
010 Please remit to our bank account which is mentioned in the announcement.
011 Please remit to our bank account which is mentioned IN THE ANNOUNCEMENT.
012 Please remit to our BANK ACCOUNT which is mentioned in the announcement.
013 The deadline is the end of the year.
014 The DEADLINE is the end of the year.
015 The deadline is the END of the year.
016 The deadline is the end of the YEAR.
017 The conference covers a wide area of research related to interpreting telephony.
018 The conference covers a WIDE AREA of research related to interpreting telephony.
019 The conference covers a wide area of research RELATED TO interpreting telephony.
020 The conference covers a wide area of research related to INTERPRETING TELEPHONY.
021 We are expecting linguists as well as psychologists as participants.
022 We are expecting LINGUISTS as well as psychologists as participants.
023 We are expecting linguists AS WELL AS psychologists as participants.
024 We are expecting linguists as well as psychologists as PARTICIPANTS.
025 Is there simultaneous interpretation into English when the presentation is made in Japanese?
026 Is there SIMULTANEOUS interpretation into English when the presentation is made in Japanese?
027 Is there simultaneous interpretation into ENGLISH when the presentation is made in Japanese?
028 Is there simultaneous interpretation into English when the presentation is made in JAPANESE?
029 By the way, what's the official language of the conference?
030 By the way, what's the OFFICIAL language of the conference?
031 By the way, what's the official LANGUAGE of the conference?
032 The conference will take place from August 22nd to 25th at the Kyoto International Conference Centre.
033 The conference will take place from AUGUST 22nd to 25th at the Kyoto International Conference Centre.
034 The conference will take place from August 22nd to 25th at the KYOTO International Conference Centre.
035 The conference will take place from August 22nd to 25th at the Kyoto INTERNATIONAL Conference Centre.
036 The conference will take place from August 22nd to 25th at the Kyoto International CONFERENCE CENTRE.
037 The fee for participation is 40,000 yen.
038 The fee for PARTICIPATION is 40,000 yen.
039 The fee for participation is 40,000 YEN.
040 Could I have your phone number too?
041 Could I have YOUR phone number too?
042 Could I have your PHONE NUMBER too?
043 I've heard that you have a city tour during the conference.
044 I've heard that you have a CITY TOUR during the conference.
045 I've heard that you have a city tour DURING the conference.
046 I've heard that you have a city tour during the CONFERENCE.
047 We'll visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the afternoon of August 5th.
048 We'll visit Kiyomizu-dera, Kinkaku-ji AND Ryoan-ji the afternoon of August 5th.
049 We'll visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the AFTERNOON of August 5th.
050 We'll visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the afternoon of AUGUST 5TH.
051 OK, please give me your name and the number of people in your party.
052 OK, please give me YOUR name and the number of people in your party.
053 OK, PLEASE give me your name and the number of people in your party.
054 OK, please give me your name AND THE NUMBER OF PEOPLE in your party.
055 Are the speakers also participating?
056 Are the SPEAKERS also participating?
057 Are the speakers ALSO participating?
058 Are the speakers also PARTICIPATING?
059 Some of them are supposed to.
060 SOME OF THEM are supposed to.
061 Some of them are SUPPOSED TO.

062 Please pay the tour fee when you arrive.
063 PLEASE pay the tour fee when you arrive.
064 Please pay the tour fee WHEN YOU ARRIVE.
065 We'll meet just in front of the reception desk.
066 We'll meet JUST IN FRONT of the reception desk.
067 We'll meet just in front of the RECEPTION DESK.
068 The titles of the papers to be presented at the conference are printed in the second version of the announcement.
069 The TITLES of the papers to be presented at the conference are printed in the second version of the announcement.
070 The titles of the papers to be presented AT THE CONFERENCE are printed in the second version of the announcement.
071 The titles of the papers to be presented at the conference are printed in the SECOND VERSION of the announcement.
072 The titles of the papers to be presented at the conference are printed in the second version of the ANNOUNCEMENT.
073 Please mail me the announcement as soon as possible.
074 Please MAIL ME the announcement as soon as possible.
075 Please mail me the announcement AS SOON AS POSSIBLE.
076 If your paper is accepted we'll enclose special forms for your paper.
077 IF your paper is accepted we'll enclose special forms for your paper.
078 If your paper IS accepted we'll enclose special forms for your paper.
079 If your paper is accepted we'll enclose SPECIAL FORMS for your paper.
080 Please take the subway to kita-oji station.
081 Please take the SUBWAY to kita-oji station.
082 Please take the subway to KITA-OJI station.
083 How much is it from Kyoto station to the conference centre by taxi?
084 How much is it from KYOTO STATION to the conference centre by taxi?
085 How much is it from Kyoto station to the conference centre BY TAXI?
086 The hotels we can help you with are the Kyoto Hotel and the Kyoto Prince Hotel.
087 The HOTELS we can help you with are the Kyoto Hotel and the Kyoto Prince Hotel.
088 The hotels WE can help you with are the Kyoto Hotel and the Kyoto Prince Hotel.
089 The hotels we can HELP YOU WITH are the Kyoto Hotel and the Kyoto Prince Hotel.
090 The hotels we can help you with are the Kyoto Hotel AND THE Kyoto Prince Hotel.
091 The hotels we can help you with are the Kyoto Hotel and the Kyoto PRINCE Hotel.
092 We'll be able to reserve rooms for you at either the Kyoto Hotel or the Kyoto Prince hotel.
093 We'll be able to reserve rooms for you at EITHER the Kyoto Hotel OR the Kyoto Prince hotel.
094 We'll be able to reserve rooms for you at either the KYOTO HOTEL or the KYOTO PRINCE HOTEL.
095 A single room will cost you from 7,000 yen to 10,000 yen per night.
096 A SINGLE room will cost you from 7,000 yen to 10,000 yen per night.
097 A single room will cost you FROM 7,000 yen TO 10,000 yen per night.
098 A single room will cost you from 7,000 yen to 10,000 yen PER NIGHT.

B Text of Database Utterances - set B

001 The proceedings AND THE RECEPTION are included in the registration fee.
002 There's no discount this time.
003 Payment should be made by BANK-TRANSFER.
004 Please remit to our bank account which is mentioned in the announcement.
005 The conference covers a WIDE AREA of research related to interpreting telephony.
006 The deadline is the end of the year.
007 Is there simultaneous interpretation into ENGLISH when the presentation is made in Japanese?
008 The conference covers a wide area of research related to interpreting telephony.
009 The deadline is the end of the YEAR.
010 Please remit to our BANK ACCOUNT which is mentioned in the announcement.
011 OK, please give me YOUR name and the number of people in your party.
012 We are expecting linguists as well as psychologists as participants.
013 The conference will take place from August 22nd to 25th at the KYOTO International Conference Centre.
014 The deadline is the END of the year.
015 Is there simultaneous interpretation into English when the presentation is made in Japanese?
016 Payment should be made by bank-transfer.
017 The conference covers a wide area of research related to INTERPRETING TELEPHONY.
018 There's NO discount this time.
019 Is there SIMULTANEOUS interpretation into English when the presentation is made in Japanese?

020 The titles of the papers to be presented at the conference are printed in the SECOND VERSION of the announcement.
021 Could I have YOUR phone number too?
022 The proceedings and the reception are included in the registration fee.
023 The DEADLINE is the end of the year.
024 By the way, what's the official language of the conference?
025 Some of them are supposed to.
026 Are the speakers also participating?
027 Is there simultaneous interpretation into English when the presentation is made in JAPANESE?
028 The hotels we can help you with are the Kyoto Hotel and the Kyoto PRINCE Hotel.
029 The conference will take place from August 22nd to 25th at the Kyoto INTERNATIONAL Conference Centre.
030 We are expecting LINGUISTS as well as psychologists as participants.
031 By the way, what's the OFFICIAL language of the conference?
032 We'll visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the AFTERNOON of August 5th.
033 I've heard that you have a city tour during the CONFERENCE.
034 The proceedings and the reception are included in the REGISTRATION fee.
035 OK, please give me your name and the number of people in your party.
036 The conference will take place from AUGUST 22nd to 25th at the Kyoto International Conference Centre.
037 I've heard that you have a CITY TOUR during the conference.
038 We'll meet JUST IN FRONT of the reception desk.
039 We are expecting linguists AS WELL AS psychologists as participants.
040 Please MAIL ME the announcement as soon as possible.
041 Could I have your PHONE NUMBER too?
042 Some of them are SUPPOSED TO.
043 A single room will cost you from 7,000 yen to 10,000 yen PER NIGHT.
044 The conference will take place from August 22nd to 25th at the Kyoto International CONFERENCE CENTRE.
045 The fee for participation is 40,000 yen.
046 We'll visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the afternoon of AUGUST 5TH.
047 By the way, what's the official LANGUAGE of the conference?
048 The fee for participation is 40,000 YEN.
049 Could I have your phone number too?
050 We'll visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the afternoon of August 5th.
051 The conference covers a wide area of research RELATED TO interpreting telephony.
052 I've heard that you have a city tour during the conference.
053 Are the speakers also PARTICIPATING?
054 The conference will take place from August 22nd to 25th at the Kyoto International Conference Centre.
055 The fee for PARTICIPATION is 40,000 yen.
056 I've heard that you have a city tour DURING the conference.
057 We are expecting linguists as well as psychologists as PARTICIPANTS.
058 We'll visit Kiyomizu-dera, Kinkaku-ji AND Ryoan-ji the afternoon of August 5th.
059 There's no discount THIS time.
060 We'll meet just in front of the reception desk.
061 OK, please give me your name AND THE NUMBER OF PEOPLE in your party.
062 Are the speakers ALSO participating?
063 We'll be able to reserve rooms for you at EITHER the Kyoto Hotel OR the Kyoto Prince hotel.
064 SOME OF THEM are supposed to.
065 The titles of the papers to be presented AT THE CONFERENCE are printed in the second version of the announcement.
066 PLEASE pay the tour fee when you arrive.
067 Please pay the tour fee WHEN YOU ARRIVE.
068 Are the SPEAKERS also participating?
069 The HOTELS we can help you with are the Kyoto Hotel and the Kyoto Prince Hotel.
070 We'll meet just in front of the RECEPTION DESK.
071 The titles of the papers to be presented at the conference are printed in the second version of the announcement.
072 Please take the SUBWAY to kita-oji station.
073 Please pay the tour fee when you arrive.
074 The titles of the papers to be presented at the conference are printed in the second version of the ANNOUNCEMENT.
075 Please mail me the announcement AS SOON AS POSSIBLE.
076 If your paper is accepted we'll enclose special forms for your paper.
077 How much is it from Kyoto station to the conference centre BY TAXI?
078 A single room will cost you FROM 7,000 yen TO 10,000 yen per night.
079 Payment SHOULD be made by bank-transfer.

080 IF your paper is accepted we'll enclose special forms for your paper.
081 If your paper IS accepted we'll enclose special forms for your paper.
082 The TITLES of the papers to be presented at the conference are printed in the second version of the announcement.
083 The hotels we can HELP YOU WITH are the Kyoto Hotel and the Kyoto Prince Hotel.
084 Please mail me the announcement as soon as possible.
085 If your paper is accepted we'll enclose SPECIAL FORMS for your paper.
086 Please take the subway to kita-oji station.
087 OK, PLEASE give me your name and the number of people in your party.
088 How much is it from Kyoto station to the conference centre by taxi?
089 The hotels we can help you with are the Kyoto Hotel and the Kyoto Prince Hotel.
090 A single room will cost you from 7,000 yen to 10,000 yen per night.
091 The hotels WE can help you with are the Kyoto Hotel and the Kyoto Prince Hotel.
092 We'll be able to reserve rooms for you at either the KYOTO HOTEL or the KYOTO PRINCE HOTEL.
093 A SINGLE room will cost you from 7,000 yen to 10,000 yen per night.
094 Please remit to our bank account which is mentioned IN THE ANNOUNCEMENT.
095 How much is it from KYOTO STATION to the conference centre by taxi?
096 The hotels we can help you with are the Kyoto Hotel AND the Kyoto Prince Hotel.
097 Please take the subway to KITA-OJI station.
098 We'll be able to reserve rooms for you at either the Kyoto Hotel or the Kyoto Prince hotel.

C Text of Database Utterances - set C

001 The proceedings and the reception are included in the application fee.
002 The proceedings AND THE RECEPTION are included in the application fee.
003 The proceedings and the reception are included in the APPLICATION fee.
004 No. There's NO discount this time.
005 No. There's no discount THIS time.
006 Payment should be made by bank-transfer.
007 Payment should be made by BANK-TRANSFER.
008 Payment SHOULD be made by bank-transfer.
009 Please remit to our bank account which is mentioned IN THE ANNOUNCEMENT.
010 Please remit to our BANK ACCOUNT which is mentioned in the announcement.
011 The DEADLINE is the end of the year.
012 The deadline is the END of the year.
013 The deadline is the end of the YEAR.
014 Well, this conference covers a wide area of research related to INTERPRETING telephony.
015 This conference covers a WIDE AREA of research related to interpreting telephony.
016 This conference covers a wide area of research related to INTERPRETING TELEPHONY.
017 We are also expecting linguists and psychologists as participants.
018 We are also expecting LINGUISTS AND PSYCHOLOGISTS as participants.
019 We are also expecting linguists and psychologists as PARTICIPANTS.
020 Is there simultaneous interpretation into English when the presentation is made in Japanese?
021 Is there SIMULTANEOUS interpretation into English when the presentation is made in Japanese?
022 Is there simultaneous interpretation into ENGLISH when the presentation is made in Japanese?
023 Is there simultaneous interpretation into English when the presentation is made in JAPANESE?
024 By the way, what's the OFFICIAL language of the conference?
025 By the way, what's the official LANGUAGE of the conference?
026 The conference will take place from August 22nd to 25th at the Kyoto International Conference Centre.
027 The conference will take place from AUGUST 22nd to 25th at the Kyoto International Conference Centre.
028 The conference will take place from August 22nd to 25th at the KYOTO International Conference Centre.
029 The conference will take place from August 22nd to 25th at the Kyoto INTERNATIONAL Conference Centre.
030 The conference (centre) will take place from August 22nd to 25th at the Kyoto International CONFERENCE CENTRE.
031 The fee for participation is 40,000 yen.
032 The fee for PARTICIPATION is 40,000 yen.
033 The fee for participation is 40,000 YEN.
034 Could I have your phone number too?
035 Could I have YOUR phone number too?
036 Could I have your PHONE NUMBER too?
037 Could I have your phone number TOO?
038 Could I have YOUR phone number too?
039 I've heard that you have a city tour during the conference.

040 I've heard that you have a CITY TOUR during the conference.
041 I've heard that you have a city tour DURING the conference.
042 I've heard that you have a city tour during the CONFERENCE.
043 We will visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the afternoon of August 5th.
044 We will visit Kiyomizu-dera, Kinkaku-ji AND Ryoan-ji the afternoon of August 5th.
045 We will visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the AFTERNOON of August 5th.
046 We will visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the afternoon of AUGUST 5TH.
047 OK, please give me your name and the number of people in your party.
048 Please give me YOUR name and the number of people in your party.
049 Please give me your name AND THE NUMBER OF PEOPLE in your party.
050 Are the speakers also participating?
051 Are the SPEAKERS also participating?
052 Are the speakers also PARTICIPATING?
053 Are the speakers ALSO participating?
054 Some of them are supposed to.
055 SOME OF THEM are supposed to.
056 Some of them are SUPPOSED TO.
057 Please pay the tour fee there when you arrive.
058 PLEASE pay the tour fee there when you arrive.
059 Please pay the tour fee there WHEN YOU ARRIVE.
060 We'll meet just in front of the reception desk.
061 We'll meet JUST in front of the reception desk.
062 We'll meet JUST IN FRONT of the reception desk.
063 We'll meet just in front of the RECEPTION DESK.
064 The titles of papers to be presented at the conference are printed in the second version of the announcement.
065 The titles of papers to be presented AT THE CONFERENCE are printed in the second version of the announcement.
066 The titles of papers to be presented at the conference are printed in the SECOND VERSION of the announcement.
067 The TITLES of papers to be presented at the conference are printed in the second version of the announcement.
068 The titles of papers to be presented at the conference are printed in the second version of the ANNOUNCEMENT.
069 Please MAIL me the announcement as soon as possible.
070 Please MAIL ME the announcement as soon as possible.
071 Please mail me the announcement AS SOON AS POSSIBLE.
072 IF your paper is accepted we'll also enclose special forms for your paper.
073 If your paper is accepted we'll also enclose SPECIAL FORMS for your paper.
074 If your paper IS accepted we'll also enclose special forms for your paper.
075 Please take the subway to kita-oji station.
076 Please take the SUBWAY to kita-oji station.
077 Please take the subway to KITA-OJI station.
078 How much is it from Kyoto station to the conference centre by taxi?
079 How much is it from Kyoto station to the conference centre BY TAXI?
080 How much is it from Kyoto station to the CONFERENCE CENTRE by taxi?
081 How much is it from KYOTO STATION to the conference centre by taxi?
082 HOW MUCH IS IT from Kyoto station to the conference centre by taxi?
083 The hotels we can help you with are the Kyoto Hotel and Kyoto Prince Hotel.
084 The hotels WE can help you with are the Kyoto Hotel and Kyoto Prince Hotel.
085 The hotels we can HELP YOU WITH are the Kyoto Hotel and Kyoto Prince Hotel.
086 We'll be able to reserve rooms for you at either the Kyoto Hotel or Kyoto Prince hotel.
087 We'll be able to reserve rooms for you at EITHER the Kyoto Hotel OR Kyoto Prince hotel.
088 We'll be able to reserve rooms for you at either the KYOTO HOTEL or KYOTO PRINCE HOTEL.
089 A single room will cost you from 7,000 to 10,000 yen per night.
090 A SINGLE room will cost you from 7,000 to 10,000 yen per night.
091 A single room will cost you FROM 7,000 TO 10,000 yen per night.
092 A single room will cost you from 7,000 to 10,000 yen PER NIGHT.

D Text of Database Utterances - set D

001 Hello. Is this the conference office?
002 Yes, that's right. May I help you?
003 I would like to attend the conference. How can I apply?
004 Please fill out a registration form. Do you have one?

005 No, not yet. (poor recording)
006 No, not yet. (poor recording)
007 I see, then I'll send you a registration form.
008 Please give me your name and address.
009 My address is 23 Chaiama-chi Kita-ku Osaka.
010 My address is 23 Chaiama-chi Kita-ku Osaka.
011 I've got it. I'll send you the form immediately.
012 Thank you very much. Goodbye.
****
013 Hello, the conference office.
014 Could you give me some information about the application fee for the conference?
015 How much will it cost if I apply for the conference right now?
016 Well let's see. It costs 35000 yen per person, but if you apply next month it will cost you 40000 yen.
017 The proceedings and the reception are included in the application fee.
018 I am a member of the Information Processing Society. Is there a discount for members?
019 No, there is no discount this time.
020 I understand. How can I pay?
021 Payment should be made by bank-transfer.
022 Please remit to our bank account which is mentioned in the announcement.
023 The deadline is the end of the year.
024 I see. Thank you very much.
025 You're welcome. Please feel free to ask if there's anything you don't understand. Goodbye.
****
026 Hello, the conference office.
027 I'd like to contribute a paper to the conference.
028 Would you please tell me the topic of the conference?
029 This conference covers a wide area of research related to interpreting telephony.
030 We're also expecting linguists and psychologists as participants.
031 I see. By the way, what is the official language of the conference?
032 English and Japanese.
033 I don't understand Japanese at all.
034 Is there simultaneous interpretation into English when the presentation is made in Japanese?
035 We have simultaneous interpretation service into English.
036 That would be helpful for me. Thank you very much. Goodbye.
****
037 The conference office.
038 I'd like to know the details of the conference.
039 Do you have an announcement of the conference?
040 No, I don't.
041 The conference will take place from August 22nd to 25th at the Kyoto International Conference Centre.
042 The fee for participation is 40000 yen.
043 If you intend to present a paper, please submit a summary by March 20th.
044 I'll send the conference announcement to you soon.
045 Would you mind telling me your name and address?
046 My name is Adam Smith.
047 My address is 2-27-7 Tamatskuri, Hiashku, Osaka.
048 Ok. Could I have your phone number too?
049 Yes. 3 7 2 - 8 0 1 8.
050 3 7 2 - 8 0 1 8.
051 Is that correct?
052 Yes it is. Thank you very much. Goodbye.
****
053 Hello, the conference office.
054 I wonder if you could help me.
055 I sent in the registration form for the conference...
056 But I can't attend the conference, so I would like to cancel.
057 Could you please give me your name?
058 This is Jim Weibel from Bell Labs.
059 You have already paid 85000 yen for your registration fee, haven't you?
060 Yes I have. Is it possible for you to refund the registration fee?
061 I'm very sorry, we can't.
062 As noted in the announcement, cancellation after September 27th precludes a refund.
063 We'll send you the programmes and proceedings later.
064 Will somebody else be able to attend the conference instead of me then?

065 That's all right. Please let me know in advance who is going to attend instead of you.
066 I understand. I'll let you know when it's decided.
****
067 Hello, the conference office.
068 I've heard that you have a city tour during the conference. Can we still take part in it?
069 Yes you can. We will visit Kiyomizu-dera, Kinkaku-ji and Ryoan-ji the afternoon of August 5th.
070 Would you like to participate?
071 How much does it cost?
072 8000 yen. That includes dinner.
073 Are the speakers also participating?
074 Some of them are supposed to.
075 I see. Then I'd also like to participate.
076 Ok, please give me your name and the number of people in your party.
077 My name is Ken Brown. My wife will also participate.
078 We'll meet just in front of the reception desk.
079 Please pay the tour fee there when you arrive.
080 I understand. Thank you very much.
081 We'll be expecting you.
****
082 Hello, the conference office.
083 I have a question about topics in the conference.
084 Yes please. What is it?
085 There's a topic called machine translation in the announcement. Specifically, what's it about?
086 I'm sorry, I'm really unable to answer any technical questions.
087 Titles of papers to be presented at the conference are printed in the second version of the announcement.
088 Would you please take a look at it?
089 Yes I will. Please mail me the announcement as soon as possible.
090 My address is 2-1-61 Shiromi, Hyashku, Osaka.
091 My name is Akira Watanabe.
092 2-1-61 Shiromi, Hyashku, Osaka.
093 Akira Watanabe, correct?
094 Yes.
095 I'll send you one as soon as possible. Is there anything else I can help you with?
096 No. Nothing more. Thank you very much. Goodbye.
****
097 Hello, the conference office.
098 Can I ask you a few questions.
099 I'd like to contribute a paper to the conference. How can I apply?
100 First you should send us a 200 word summary by March 20th.
101 The summary will be reviewed here.
102 And we'll send you a reply by May 20th.
103 If your paper is accepted, we'll also enclose special forms for your paper.
104 Please send them back to us by June 30th.
105 I understand. What kind of form do I have to write the summary on?
106 We have a special form for the summary. Please fill it in, then we'll send you the application form.
107 May I have your name and address please?
108 Alright, my name is George O'Hara from AI Labs.
109 My address is 3-2-5 Hiyashi, Ikevukero, Toshimaku, Tokyo.
110 Mister George O'Hara from AI Labs, right?
111 Your address is 3-2-5 Hiyashi, Ikevukero, Toshimaku, Tokyo. Is that correct?
112 Yes, it is. Please send me an application form.
113 Yes, I will. I'll send it to you immediately. Goodbye.
****
114 Hello, the conference office.
115 Can I ask you a few questions?
116 I'd like to contribute a paper to the conference. How can I apply?
117 First you should send us a 200 word summary by March 20th.
118 The summary will be reviewed here.
119 And we'll send you a reply by May 20th.
120 If your paper is accepted, we'll also enclose special forms for your paper.
121 Please send them back to us by June 30th.
122 I understand. What kind of form do I have to write the summary on?
123 We have a special form for the summary. Please fill it in, then we'll send you the application form.
124 May I have your name and address please?

125 Alright, my name is George O'Hara from AI Labs.
126 My address is 3-2-5 Hiyashi, Ikevukero, Toshimaku, Tokyo.
127 Mister George O'Hara from AI Labs, right?
128 Your address is 3-2-5 Hiyashi, Ikevukero, Toshimaku, Tokyo. Is that correct?
129 Yes, it is. Please send me an application form.
130 Yes, I will. I'll send it to you immediately. Goodbye.
****
131 Is this the conference office?
132 Yes, this is the conference office. May I help you?
133 Please tell me how to get to the conference site. I'm now at Kyoto station.
134 Please take the subway to Kita-oji station. From there, there is a bus to the conference centre.
135 Of course, you'll also be able to take a taxi from Kita-oji station.
136 How much is it from Kyoto station to the conference centre by taxi?
137 From Kyoto station, it'll cost you about 6000 yen.
138 And how much does it cost from Kita-oji station?
139 From Kita-oji station, it'll cost you approximately 900 yen.
140 I see. Thank you very much.
141 Not at all. You're welcome.
****
142 Hello, the conference office.
143 I'd like to ask you about hotel accommodations for the conference.
144 Do you have a service that will help me find a place to stay?
145 Yes we do. The hotels we can help you with are the Kyoto Hotel and Kyoto Prince Hotel.
146 A single room will cost you from 7000 to 10000 yen per night.
147 A twin room ranges from 9500 to 60000 yen.
148 I see. Which hotel is closer to the conference centre?
149 The Kyoto Prince Hotel is closer to the conference centre.
150 Then I'd like to make a reservation for the Kyoto Prince Hotel. Can I leave the hotel reservation to you?
151 Sure. We'll be able to reserve rooms for you at either the Kyoto Hotel or the Kyoto Prince Hotel.
152 That's fine. Well, could you reserve a 7000 yen single room at the Kyoto Prince Hotel?
153 Ok. A 700 yen single room at the Kyoto Prince Hotel, right?
154 Yes, that's right.
155 When will you check in?
156 The evening of August 4th. Checking out the morning of the 8th.
157 Ok. Please wait a moment. I'm going to check to see if there's a vacancy.
158 Yes there is. Please give me your name and address.
159 My name is Kazuo Nakamura.
160 The address is 1-1-3 Shimbashi, Minatoku, Tokyo.
161 And your phone number please?
162 And your phone number please?
163 My phone number is 3 3 1 - 2 5 2 1.
164 Ok. I have reserved a single room at the Kyoto Prince Hotel from August 4th to 8th.
165 Thank you very much. Goodbye.

E PCB user commands: manual pages

adlb_hditem (PCB)	— print an item from an Audlab file header
adlb_xmg (PCB)	— convert Audlab headed track and label files to Xmg format
alter_adlb_hd (PCB)	— alter Audlab file headers
aps_syl (PCB)	— make a trivial Audlab label file for APS
auto_prosody (PCB)	— automatic prosodic transcription
binary2ascii (PCB)	— binary to ascii file conversion
cep_fft (PCB)	— fast Fourier transform based (improved) cepstral analysis
cep_syn (PCB)	— cepstral speech synthesis
cep_split (PCB)	— split joint pitch and cepstra file
cep_stitch (PCB)	— stitch separate pitch and cepstra files
compf0 (PCB)	— compare files describing F0 contours in Xmg format
diac2label (PCB)	— form label files from the diacritics of a label file
dur (PCB)	— determine total unvoiced and voiced durations from Xmg F0 contours
energy (PCB)	— calculate normalised energies within specified frequency bands
f0_hist (PCB)	— histogram, mean and standard deviation of Xmg F0 contours
gcipd (PCB)	— a pitch determination algorithm based on glottal closure instants
getminload (PCB)	— get name of host with minimum load
getpid (PCB)	— get process ID for a given name from a file
hps (PCB)	— a pitch determination algorithm based on the harmonic product spectrum
labelformat (PCB)	— convert between Segstat, HMM, Audlab, and Waves label file formats
mix_stats (PCB)	— accumulate F0 contour comparison statistics
normf0 (PCB)	— declination compensation and normalisation of F0 contours
pc2adlb_vox (PCB)	— convert PC speech files to Audlab headed format
piecewise (PCB)	— an F0 contour piecewise stylisation algorithm
pm_freq (PCB)	— pitch mark data to Xmg-format F0 contour
pmlar (PCB)	— a pitch marker for laryngograph data
prosody_stats (PCB)	— calculate normalisation statistics for prosodic analysis
read_adlb_hd (PCB)	— read header of Audlab file
rm_adlb_hd (PCB)	— remove header from Audlab file
sdt2trk (PCB)	— convert Audlab spatial domain track file to a simple track file
smoother (PCB)	— a non-linear smoothing algorithm for Audlab parameter tracks and sample data
spectral_inversion (PCB)	— invert the spectral characteristics of a speech waveform
srpd (PCB)	— a super resolution pitch determination algorithm (Sun Version)
xmg_adlb (PCB)	— convert a Xmg format segment file to an Audlab headed track file

NAME

adlb_hditem – print an item from an Audlab file header

SYNOPSIS

adlb_hditem [-s] [-t **track/channel**] -i *item_name audlab_file*

DESCRIPTION

An item from an Audlab file header is printed on the standard output. If the [–s] (silent) option is given, it merely returns a success (0) or failure status (1), depending on whether or not the given *item_name* is defined in the file header.

The following "item names" are currently recognised:

Display Defaults

lower_limit
: lowest data value

upper_limit
: highest data value

break_number
: code to break line when data is displayed

plot_style
: style in which to plot data – returns "DONTPLOT", "POINTS", "LINES" or "UNDEFINED"

dash_dot
: style in which to draw lines - returns "DASH", "DOT", or "UNDEFINED"

linewidth
: graphics line widith

hue
: colour in which to draw data - returns "red", "green", "blue", "yellow", "orange", "LightRed", "LightGreen", "LightBlue", "white", or "black"

File Header

name
: file segment identification name

start_time
: file segment start time in seconds

stop_time
: file segment stop time in seconds

filetype
: type of data stored in file - returns "Sample", "Track", "FFT", "SDT", or "Label"

filedescr description of the file

hist_file name of history file

parent file name of parent segment

start_of_data
offset from the beginning of the file to the start of the data in bytes

data_fmt
format in which data is stored - returns "FLOAT", "INT", "SHORT", "SSL", or "UNDEFINED"

spare any addition details

Sample Header

nchannels
number of channels

par_serial
channel storage style - returns "PARALLEL", "SERIAL", or "UNDEFINED"

sample_freq
data sampling frequency in Hz

speaker_id
speaker reference

age age of speaker

sex sex of speaker - returns "MALE", "FEMALE", or "UNDEFINED"

mike microphone details

ambience
ambient recording conditions

Sample Descriptor Header

descr description of channel

chanlen number of data elements in channel

nbits analogue-to-digital convertor resolution in bits

aacutoff anti-aliasing filter cut-off frequency in Hz

Fast Fourier Transform (FFT) Header

descr description of transform

nchannels
number of frequency bands

fftsize number of analysis points

sample_freq
data sampling frequency in Hz

frame_length
analysis interval in seconds

frame_shift
shift of interval in seconds

tracklen number of data elements

Spatial Domain Track (SDT) Header

descr description of transform

nchannels
number of frequency bands/channels

sample_freq
data sampling frequency in Hz

chn_min display value of first channel

chn_max
display value of last channel

frame_length
analysis interval in seconds

frame_shift
shift of interval in seconds

tracklen number of analysis frames

xdomain
x–axis label

ydomain
y–axis label

Label Header

method method used in labelling - returns "HAND", "AUTO", or "UNDEFINED"

type type of labels stored - returns "ORTHO", "BROAD", "MIDDLE", "FINE", "PHONEMIC", "PHONETIC", or "UNDEFINED"

labeller labeller identification (name of person or program)

Track Header

ntracks number of tracks

par_serial
track storage style - returns "PARALLEL", "SERIAL", or "UNDEFINED"

track_dimensions
dimensions of track values - returns "Y_DIM", "XY_DIM", "XYZ_DIM", or "UNDEFINED"

sample_freq
data sampling frequency in Hz

frame_length
analysis interval in seconds

frame_shift
shift of interval in seconds

Track Descriptor Header

descr description of track

tracklen number of data elements in track

OPTIONS

−s silent mode. Just return success (0) or failure (1) to the shell (unless an error occurs in which case (-1) is returned).

−t *track/channel* read Audlab track or channel header for the specified *track/channel* in a multiple header feature or speech file respectively. The default is to search for the item named in the first (1) track/channel header.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

adlb_xmg – convert Audlab headed track and label files to Xmg format

SYNOPSIS

adlb_xmg -i *track file* -o *xmg track*

[–t *track number*]

[–s *y scale multiplier*]

[–l *xmg track lower limit*]

[–u *xmg track upper limit*]

[–c *colour name*]

adlb_xmg -i *label file* -o1 *xmg bars*-o2 *xmg labels*

[–c *colour name*]

[–f *sampling frequency*]

[–h *height of labels*]

[–d *diacritic symbols file*]

DESCRIPTION

The track given by [–t] *track number* in [–i] *track file* is transferred into [–o] *xmg track* with an Xmg header. The [–o] *xmg track* will contain lists of points with each list separated by an '=' character. Each list of points are to be drawn in Xmg with solid interconnecting lines in the same colour as when the track is drawn using Audlab; unless the [–c] *colour name* is passed, then it is to be drawn in the colour specified. Each element of the Audlab track is multipled by [–s] *y scale multiplier* before being entered into [–o] *xmg track*. This scale may be used to enable Audlab data in the range 0 to 1 to be plotted with Xmg. The minimum and maximum y–axis values are inherited from the Audlab header, but may be specified using [–l] *xmg track lower limit* and [–u] *xmg track upper limit* respectively.

The segment times and labels given in [–i] *label file* are converted into [–o1] *xmg bars* and [–o2] *xmg labels* respectively. The name of the colour in which the bars and labels are to be drawn in Xmg is specified by [–c] *colour name*. If a colour is not named, bars and labels are coloured blue by default. (This becomes black on a monochrome screen.) The sampling frequency field of the Xmg header is set by [–f] *sampling frequency*. The y–axis position of the labels is given by [–h] *height of labels*. The label characters are inherited from the Audlab file description, but if the label is a null string, the output Xmg file label contains a single hash (#). Diacritic codes are converted into ascii symbols using [–d] *diacritic symbols file*. The label string and diacritic symbols are separated by an underscore character (_). The x–axis position of the label and diacritic symbols (if any) is given by the start time of each segment descriptor. The boundary time (bar position) between one segment and its successor is given by the stop

time of the segment descriptor.

OPTIONS

–t *track number* specifies that the *track number*'th track of [–i] *track file* is to be converted to Xmg format. The default value is one (the first track). Tracks may be in parallel or serial format.

–s *y scale multiplier* set the multiplication factor which is applied to the input data to *y scale multiplier*. This must be an integer value and has the default of one.

–l *xmg track lower limit* the lower y–axis limit on the Xmg display window. The default value is read from the Audlab header.

–u *xmg track upper limit* the upper y–axis limit on the Xmg display window. The default value is read from the Audlab header.

–c *colour name* set the colour of the track, bars, or labels when drawn using Xmg to *colour name*. The default colour is read from the Audlab header for track input files, and blue for label input files.

–f *sampling frequency* specifies the sampling frequency value for the Xmg header of label and boundary files as *sampling frequency* in kHz. Default is 20kHz.

–h *height of labels* set the y–axis coordinate of the label positions to *height of labels*. The default is zero.

–d *diacritic symbols file* specifies the location (full path name) and name of the Audlab format, diacritic symbols description file. The default is /u10/audlab/v3.0/public/MEN/diacrit.sym.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.
XMG, a multiple graph drawer for X11.

DIAGNOSTICS

The diagnostics produced by ADLB_XMG are intended to be self–explanatory.

BUGS

The type of Audlab formats which may be converted is limited to track and label files. The Xmg files created may cause bugs in the Xmg program to show up - tracks with negative values and labels with height zero on the same window can be a problem to Xmg.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

alter_adlb_hd – alter Audlab file headers

SYNOPSIS

alter_adlb_hd *audlab file*

[**–d** *start of data*]
[**–t** *channel/track number*]
[**–f** *sampling frequency*]
[**–l** *lower limit*]
[**–u** *upper limit*]
[**–b** *break number*]
[**–c** *colour name*]
[**–w** *frame length*]
[**–s** *frame shift*]
[**–n** *channel ADC resolution*]
[**–m** *label method*]
[**–p** *label type*]

DESCRIPTION

The AUDLAB file header descriptors are altered in accordance with the values passed by the optional parameters. The new values selected over-write those previously in the file's headers. A header value is left unaltered if the corresponding parameter is omitted from the command line. The *audlab file* may be of sample, track, fft, sdt, or label file type. The options that can be passed to the program depends upon the type of the input file.

The input file types and corresponding possible options are:

sample [–d] [–t] [–f] [–l] [–u] [–b] [–c] [–n]

track [–d] [–t] [–f] [–l] [–u] [–b] [–c] [–w] [–s]

fft [–d] [–f] [–l] [–u] [–b] [–c] [–w] [–s]

sdt [–d] [–f] [–l] [–u] [–b] [–c] [–w] [–s]

label [–d] [–m] [–p]

OPTIONS

–d *start of data* set the field specifying the location (in bytes) of the first data point from the beginning of the file to *start of data.*

–t *channel/track number* specify that the channel of sample data or track whose descriptor is to be altered . The default is channel/track one (the first in the file).

–f *sampling frequency* change the sampling frequency description to the value *sampling frequency.* The value is specified in Hertz.

–l *lower limit* set the default display lower data limit to *lower limit.*

–u *upper limit* set the default display upper data limit to *upper limit.*

–b *break number* set the breaker value to *break number.*

–c *colour name* set the colour in which the data is to be drawn by audlab to *colour name.* The colours that may be chosen are red, green, blue, yellow, orange, LightRed, LightGreen, LightBlue, and white.

–w *frame length* set the frame length descriptor to the value *frame length* (seconds).

–s *frame shift* set the frame shift descriptor to the value *frame shift* (seconds).

–n *channel ADC resolution* set the analogue-to-digital converter resolution description for the selected channel of sample data to *channel ADC resolution* (bits).

–m *label method* set the flag describing the method of labelling used to either *hand* or *auto.*

–p *label type* set the flag describing the type of labels to either *ortho* (orthographic labelling), *broad* (broad class), *middle* (middle class), *fine* (fine class), *phonemic* (MRPA symbols), or *phonetic* (phonetic symbols).

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

DIAGNOSTICS

The diagnostics produced by ALTER_ADLB_HD are intended to be self–explanatory.

BUGS

The program alters only a subset of the possible Audlab header fields.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

aps_syl – make a trivial AUDLAB label file for APS.

SYNOPSIS

aps_syl –**i** *audlab_data_file* –**o** *label_file*

DESCRIPTION

The program performs the trivial act of forming the AUDLAB label file [–o] *label_file*, containing a single labelled segment with a start time and a stop time that correspond exactly with the times specified in the header of the speech file [–i] *audlab_data_file*. The label applied is the segment name given in the speech file header. This enables APS to use all track data that may have been calculated for an utterance in the case when a label transcription file does not contain labels for the entire duration of the utterance.

OPTIONS

None.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

J.M. Harrington, and G.S. Watson, "A.P.S. User Guide: An Environment for Acoustic Phonetic Research," CSTR., University of Edinburgh.

DIAGNOSTICS

The diagnostics produced by APS_SYL are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

auto_prosody – automatic prosodic transcription

SYNOPSIS

auto_prosody –p *parameters file*
–s *statistics file*
–f *features file*
–l *segmentation file*
–o *prosody labels file*
[–a *analysis output stage*]

DESCRIPTION

The algorithm implemented automatically transcribes prosodic events in a speech utterance using the approach described in (Bagshaw, 1992). All variable parameters and database/speaker dependent statistics are read from ASCII files which must be written in a format specific to this **prosody**(PCB) software. These variable are given by [–p] *parameters file* and [–s] *statistics file* respectively. The transcription program requires two files to describe the phonemic content of the utterance ([–l] *segmentation file*) and acoustic features of the speech waveform ([–f] *features file*). These are used to obtain a syllabification of the utterance and a series of intonation units. Sentential stress and pitch movements derived from such units are related to each syllable identified. The prosodic transcription formed is written to the [–o] *prosody label file*. The optional [–a] *analysis output stage* specifies the type of information to be written into the prosody label file.

Parameters File

The [–p] *parameters file* is required to contain the following information given in the format illustrated. Lines beginning with a hash (#) are ignored and may include comments.

Label strings that occur in the input segmentation file are categorised into one of four groups; namely syllable nuclei, syllabic consonants, pause labels, and others. If any of the possible labels falls into one of the first three of these categories, it must be specified as a member of a label list following one of the keywords **syl_nuclei**, **syllabic_consonants**, or **pause_labels**, respectively. For the Edinburgh University MRPA (machine readable phonetic alphabet), this is specified as:

:start label_list
syl_nuclei ii i e a ah @ @@ @r uh uu u oo o aa ei ai oi ou au i@ e@ u@
syllabic_consonants l m n r .
pause_labels ## # + %

:end

Any number of label strings may follow each keyword. Each label is separated by a white space and the label list is terminated by a newline character (\n).

The labels used in the output file to describe varies prosodic events must be specified. There are no default values and each label must be different.

:start labels
max_duration Sd
dur_stressed sd
dur_unstressed ud
max_energy Se
nrg_stressed se
nrg_unstressed ue
max_pitch Sp
pitch_stressed sp
pitch_unstressed up
unknown ?
level l
rise r
fall f
risefall ^
fallrise v
accented a
stressed s
unstressed u
:end

The normalised energy track (see **energy**(PCB)) and smoothed fundamental frequency track (see **srpd**(PCB)) numbers of the input features file (named by the [–f] option) must be specified. Numbers specifying the tracks containing confidence weights for the F0 values and a stylised F0 contour (see **piecewise**(PCB)) may also be set. The default numbers for the **f0_track**, **weight_track**, **energy_track**, and **styleF0_track** are 1, 2, 3 and 4 respectively. Setting the weight_track to zero is interpreted as F0 confidence weights not being available. The stylef0_track can be set to zero to indicate that stylisation is to be performed during the prosodic analysis rather than having already been determined using **piecewise**(PCB).

:start int_params
f0_track 2
weight_track 0
energy_track 4
stylef0_track 5
:end

The amount of dither to be tolerated in the energy contour so as not to falsely identify local minima during the syllabification stage, may be specified by using the

keyword **energy_differ**. The default value is 0.0dB.

:start float_params
energy_dither 0.0
:end

Two thresholds are used in categorising a syllable as stressed on the grounds of duration and energy. If the smoothed, normalised duration or energy is above the respective threshold, and it is a local maximum in the contour, the syllable is potentially prominent in the utterance. The default value is 1.0 for both thresholds. They are specified in terms of number of standard deviations from the mean, using the following format:

:start float_params
duration_threshold 0.0
energy_threshold 0.0
:end

Five parameters are set to describe how the stylised F0 contour (in track **stylef0_track** of the features file) was generated (using **piecewise**(PCB)) or how a temporary contour is to be created during forming the prosodic transcription. A temporary stylised F0 contour is produced if the **stylef0_track** is set to zero. The function of each of these parameters is analogous to those described for **piecewise**(PCB). The parameters may be set using the following format (default values shown):

:start int_params
lsfit_method 0
init_window_len 5
hertz_out 1
:end
:start float_params
mindur 100.0
maxvar 1.0
:end

A number of parameters are also used when relating piecewise units to pitch movements. Peaks and valleys in the piecewise contour are located with an F0 tolerance specified by the **f0_dither** keyword (default is 0.0Hz). Least median of squared residuals regression analysis is performed on the peaks, valleys and all the piecewise contour turning points to give top, base and mid lines respectively. The standard deviation of the points used in the analysis from the resultant model are weighted by three variables. They are set using the keywords **top_sd_spread**, **base_sd_spread**, and **mid_sd_spread** (default values are 2.5, 2.5, and 1.5 respectively). Three parameters may be set for use in determining the pitch prominence of a given syllable. They are, the F0 step from one piecewise unit to the next, **pitch_delta**, and the F0 slope in Hertz per second of an F0 rise and an F0 fall, **inc_slope** and **dec_slope**

(default values are 9.0Hz, 100.0Hz/s, and -100Hz/s respectively). The latter two are currently not used. These parameters may be set using the format below.

:start float_params
f0_dither 0.0
top_sd_spread 1.5
base_sd_spread 1.5
mid_sd_spread 1.0
pitch_delta 9.0
inc_slope 100.0
dec_slope -100.0
:end

Speaker Dependent Statistics File.

The [-s] *statistics file* gives, for a particular speaker, the mean and population standard deviation (p.s.d.) of the fundamental frequency for all voiced frames in a training database. The mean and p.s.d. duration and energy for every possible phone type must also be given. If a phone type is encounted for which these statistics are unknown, a warning is be given. Currently, the **mean_f0** and **sd_f0** parameters are not used. These statistics can be gathered from a training database and presented in the necessary format by using **prosody_stats**(PCB).

Acoustic Features File.

The [-f] *features file* specifies the name of an Audlab track file containing acoustic features for the utterance. The fundamental frequency contour and sonorant energy contour of the utterance must be presented in the acoustic features file. Optional tracks can describe a stylised F0 contour and a weight of confidence for each F0 value. The length of these two optional tracks must be the same as that of the fundamental frequency contour. The track numbers of the acoustic features must be specified in the parameters file.

Phonemic Segmentation File.

The [-l] *segmentation file* gives the name of an Audlab label file containing a phonemic transcription of the utterance. The program **labelformat**(PCB) can be used to transform some transcription file formats to the required Audlab format. The labels that are possible in this file must be specified in the parameters file and statistics must exist for each phone type in the speaker dependent statistics file.

Prosody Labels File.

The output of the prosodic transcription program is placed in the file [-o] *prosody labels file*. The format of this file and its contents is dependent upon the setting of [-a] *analysis output stage*.

If the output stage is selected as either **syllable**, **duration**, **energy**, **pitch**, or **combined**, the output file is an Audlab label file. In all cases, the start and stop times of each segment correspond to syllable boundaries. No diacritics are

included. The label field is set to the syllable nucleus label if the setting **syllable** is selected. If the setting is **duration**, **energy**, or **pitch**, then the label field indicates if the syllable is prominent on the grounds of duration, energy or pitch, respectively. The labels used correspond to maximum duration/energy/pitch in the utterance (*max_duration*, *max_energy*, *max_pitch*), prominent on the basis of duration/energy/pitch (*dur_stressed*, *nrg_stressed*, *pitch_stressed*), and not prominent by duration/energy/pitch (*dur_unstressed*, *nrg_unstressed*, *pitch_unstressed*). If the setting is **combined**, the label field indicates the prominence of the syllable as *accented*, *stressed*, or *unstressed*. The labels are specified in the parameters file.

If the output stage **general** is selected, the output file is an ASCII file containing 14 fields as follows.

1. syllable nucleus label.
2. syllable start time in milliseconds.
3. syllable stop time in milliseconds.
4. duration measure - maximum, smoothed, normalised phone duration in syllable.
5. time location in milliseconds of the maximum, smoothed, normalised phone energy in syllable.
6. energy measure - maximum, smoothed, normalised phone energy in syllable.
7. Shape of pitch movement associated with syllable - using the labels *unknown*, *level*, *rise*, *fall*, *risefall*, and *fallrise* specified in the parameters file.
8. Relative pitch height at left hand side of pitch movement.
9. Relative pitch height at right hand side of pitch movement.
10. Relative pitch height at mid-point of pitch movement in the case of a rise-fall and fall-rise. This is set to zero for other types of movement.
11. Level for stress from combined measures.
12. Prominence on the basis of duration label.
13. Prominence on the basis of energy label.
14. Prominence on the basis of pitch label.

OPTIONS

–p *parameters file*	specifies the name of the ASCII parameters file stipulating the many possible variables for the prosodic transcription. The format of this must comply with the example descriptions given above.
–s *statistics file*	specifies the file name whose contents describe phone dependent and speaker dependent statistics. The file may be generated automatically by using the program **prosody_stats**.
–f *features file*	an Audlab track file contain at least two tracks to describe the fundamental frequency and the energy contour for an utterance.
–l *segmentation file*	an Audlab label file containing a phonemic transcription of

the speech utterance.

–o *prosody labels file* specifies the name of the output file of the prosody transcription file. Th information and format of this file is dependent upon the analysis output stage set using the [–a] flag.

–a *analysis output stage* the type of information to be presented in the output file can be selected by using this flag. Possible settings are **syllable, duration, energy, pitch, combined, general**. The default setting is **combined**. The interpretation of each of these possible stages is described above.

SEE ALSO

energy (PCB) – calculate normalised energies within specified frequency bands.
labelformat (PCB) – convert between Segstat, HMM, Audlab, and Waves label file formats.
piecewise (PCB) – an F0 contour piecewise stylisation algorithm.
prosody_stats (PCB) – calculate normalisation statistics for prosodic analysis.
srpd (PCB) – a super resolution pitch determination algorithm.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

P.C. Bagshaw, "An investigation of acoustic events related to sentential stress and pitch movements, in English," Proceedings of the 4th Australian International Conference on Speech Science and Technology, Brisbane (1992).

DIAGNOSTICS

The diagnostics produced by AUTO_PROSODY are fully comprehensive and intended to be self–explanatory.

BUGS

None known. However, the algorithm implementation has changed frequency, so bugs are highly likely.

AUTHOR

Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai

Seika-cho
Kyoto 619-02
Japan.

NAME

binary2ascii – binary to ascii file conversion

SYNOPSIS

binary2ascii [–s|–i|–f|–d] *binary file* [*ascii file*]

DESCRIPTION

The data in file *binary file* must be of type 'short', 'integer', 'float', or 'double'. The type of data known to exist in the input file is specified by the flag [–s(hort)|–i(nteger)|–f(loat)|–d(ouble)]. The default is [–d(ouble)]. It is converted to ascii format with one value per line (each line terminated with '\n'). The data in ascii format is written to the file *ascii file*, if specified; otherwise the output is directed to the file stream *stdout*.

DIAGNOSTICS

The diagnostics produced by the BINARY2ASCII program inform the user of any problems in creating or reading files, or of misuse at the command line.

BUGS

None Known.

AUTHOR

Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai
Seika-cho
Kyoto 619-02
Japan.

NAME

cep_fft – fast Fourier transform based (improved) cepstral analysis

SYNOPSIS

cep_fft **–i** *speech file* **–o** *cepstra file*
[**–n** *cepstrum order*]
[**–p** *power of 2 Fourier transform size*]
[**–f** *sampling frequency*]
[**–s** *frame shift*]
[**–w** *frame length*]
[**–r** *number of improvement iterations*]
[**–a** *speed–up convergence factor*]
[**–z**]

DESCRIPTION

This program generates a file of (improved) cepstral coefficients (including the zeroth coefficient) for speech data using fast Fourier transforms.

Segments of data of length [–w] *frame length* (ms) are taken from the speech data in [–i] *speech file*. The speech data is assumed to have been sampled at [–f] *sampling frequency* (Hz). Successive frames are shifted by [–s] *frame shift* (ms). Each frame is passed through a Blackman window and its logarithmic power spectrum is calculated by a fast Fourier transform using (2 to the power of [–p] *power of 2 transform size*)-points. The FFT size must be big enough to contain all the samples in a frame. Zero padding is used to fill excess points. If the [-z] flag is passed the zeroth bin (d.c.) of this log power spectrum is set to zero. An inverse FFT is performed on the resultant spectrum to form a cepstrum. The cepstrum is improved using the technique described in (Imai and Abe, 1979). The first [–n] *cepstrum order* coefficients plus one (the zeroth cepstrum coefficient) of each frame are written to the file [–o] *cepstra file* in binary format.

The smoothed estimate of the log power spectrum that is generated from [–n] *cepstral order* (plus one) non-improved coefficients, does not form the envelope of the peaks of the original log power spectrum. Therefore the variation from cepstral regenerated spectrum of one frame to the next is not smooth. (Imai and Abe, 1979) propose an iterative method that adjusts the cepstral coefficients such that the smoothed spectrum forms an envelope of the log power spectrum peaks. The half-wave rectified error between the smoothed spectrum and the log power spectrum is determined. This error is transformed to the cepstral domain, processed by a symmetric Hamming window of order [–n] *cepstral order*, scaled by a factor one plus [–a] *speed–up convergence factor*, and added to the cepstral coefficients. The number of times this process is repeated is given by [–r] *number of improvement iterations*.

OPTIONS

–i *speech file* specifies *speech file* as input to the algorithm. Must be an headerless file containing sample data of type 'short'.

–o *cepstra file* specifies *cepstra file* as output from the algorithm. This will be a headerless file containing parallel structures of 'float' type cepstral values for each frame analysed.

–n *cepstrum order* specifies the number of cepstral coefficients required as *cepstrum order*, excluding the zeroth coefficient which is always given. The maximum number of coefficients that can be made is dependent upon the size of the FFT used. The default is 30 (plus one) cepstral coefficients.

–p *power of 2 Fourier transform size* set the number of points to be used in the FFT to two to the power of *power of 2 Fourier transform size*. Must be greater than five. Default setting is 8 (a 256–point FFT).

–f *sampling frequency* specifies the rate at which the original speech was sampled at as *sampling frequency*, in Hertz. Default is 12000Hz.

–s *frame shift* set the shift between consecutive analysis frames to *frame shift* in milliseconds. Must be greater than zero. Default frame shift is 5.0ms.

–w *frame length* set the length of the analysis frame to *frame length* in milliseconds. Must be greater than zero. Default frame length is 21.3ms.

–r *number of improvement iterations* gives the number of times the cepstral coefficient improvement process is to be repeated as *number of improvement iterations*. The value given must be positive. The default is 5.

–a *speed–up convergence factor* gives the smoothed error scaling factor used to speed–up the convergence of the smoothed spectrum with the envelope of log power spectrum peaks as *speed–up convergence factor*. The value given must be greater than or equal to zero. The default setting is 1.0.

–z set the zeroth bin of the FFT to zero for each frame analysed before calculating the cepstrum. This is included in an attempt to remove unwanted d.c. ripple in the input speech signal.

SEE ALSO

cep_syn (PCB) – cepstral speech synthesis.
cepf0_split (PCB) – split joint pitch and cepstra file.
cepf0_stitch (PCB) – stitch separate pitch and cepstra files.

DOCUMENTATION

S. Imai and Y. Abe, "Spectral envelope extraction by improved cepstral method," Trans. IEICE Japan Vol.J62-A No.4 pp.217-223 (1979) (in Japanese).

BUGS

None known.

AUTHORS

K Abe and Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai
Seika-cho
Kyoto 619-02
Japan.

NAME

cep_syn – cepstral speech synthesis

SYNOPSIS

cep_syn **–p** *pitch file* **–c** *cepstra file* **–o** *synth file*
[**–n** *cepstrum order*]
[**–f** *sampling frequency*]
[**–s** *frame shift*]
[**–a** *amplitude scaler*]
[**–b** *break number*]

DESCRIPTION

This program synthesises speech from cepstral coefficients and a fundamental frequency contour.

OPTIONS

–p *pitch file* specifies *pitch file* as the file containing a series of F0 values, one for each frame. It must be a headerless file containing F0 values of type 'float'.

–c *cepstra file* specifies *cepstra file* as the file containing the cepstral coefficients. This must be a headerless file containing parallel structures of *cepstrum order* + 1 'float' type cepstral values listed for each frame in series.

–o *synth file* specifies *speech file* as input to the algorithm. Must be an headerless file containing sample data of type 'short'.

–n *cepstrum order* specifies the number of cepstral coefficients given in *cepstra file* for each frame as *cepstrum order*, excluding the zeroth coefficient which must always be given. The default is 30 (plus one) cepstral coefficients.

–f *sampling frequency* specifies the rate at which the synthesised speech is sampled at as *sampling frequency*, in Hertz. Default is 12000Hz.

–s *frame shift* set the time shift between frames to *frame shift* in milliseconds. Must be greater than zero. Default frame shift is 5.0ms.

–a *amplitude scaler* set the factor by which each sample value of the synthesised speech is multiplied by to prevent clipping as *amplitude scaler*. The default scaling factor is 0.5.

–b *break number* breaks in the F0 contour described in the file *pitch file* which represent unvoiced speech are given by the value *break number*. The default setting is 0.0.

SEE ALSO

cep_fft (PCB) – fast Fourier transform based (improved) cepstral analysis.
cepf0_split (PCB) – split joint pitch and cepstra file.
cepf0_stitch (PCB) – stitch separate pitch and cepstra files.

DOCUMENTATION

S. Imai, "Log magnitude approximation (LMA) filter," Trans. IEICE Japan Vol.J63-A No.12 pp.886-893 (1980) (in Japanese).

BUGS

The program does not make use of 'malloc' and so segmentation errors may occur at run time for some parameter selections. The synthesis is currently hard wired for 30 (plus one) cepstral coefficients (shame really). Data should therefore be down–sampled to 12kHz prior to cepstal analysis. Diagnostics are limited - so take care with parameter selection.

AUTHORS

K Abe and Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai
Seika-cho
Kyoto 619-02
Japan.

NAME

cepf0_split – split joint pitch and cepstra file

SYNOPSIS

cepf0_split –i *pitcep file* –p *pitch file* –c *cepstra file*
[–n *cepstrum order*]
[–f *sampling frequency*]
[–b *break number*]

DESCRIPTION

The pitch and cepstra information given in the single file [–i] *pitcep file* are split into the two separate files [–p] *pitch file* and [–c] *cepstra file* respectively. The *pitcep file* is assumed to contain binary data of type 'float'. The first value is taken as a pitch period in units of number of samples (the sampling rate given by [–f] *sampling frequency*). A pitch period value of zero represents unvoiced speech. If the pitch period in number of samples represents voiced speech, it is converted to a pitch frequency in Hertz and written to the *pitch file*; otherwise it is represented by [–b] *break number* before being written. The next [–n] *cepstrum order* plus one (for the zeroth cepstral coefficient) values in the *pitcep file* are taken to be cepstral coefficients and are transferred to the *cepstra file* without any conversion. This pattern of pitch value followed by a series of cepstral coefficient values is assumed to repeat through the input file an integer number of times. Data is transferred each time in the manner described above.

OPTIONS

–i *pitcep file* specifies *pitcep file* as the file containing the pitch and cepstral coefficients combined. Must be a headerless file containing data of type 'float'.

–p *pitch file* specified *pitch file* as the headerless file to be created with pitch values in Hertz written in type 'float'.

–c *cepstra file* specifies *cepstra file* as the headerless file to be created with parallel structures of 'float' type cepstral values.

–n *cepstrum order* specifies the number of cepstral coefficients given in the input file as *cepstrum order*, excluding the zeroth coefficient which must always be given. The default is 30 (plus one) cepstral coefficients.

–f *sampling frequency* specifies the sampling rate in which terms the pitch period is given, as *sampling frequency*, **in Hertz. Default is 12000Hz.**

–b *break number* **breaks in the F0 contour described in the file** *pitch file* **which represent unvoiced speech are given by the value** *break number*. **The default setting is 0.0.**

SEE ALSO

cep_fft (PCB) – fast Fourier transform based (improved) cepstral analysis.
cep_syn (PCB) – cepstral speech synthesis.
cepf0_stitch (PCB) – stitch separate pitch and cepstra files.

BUGS

None known.

AUTHOR

Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai
Seika-cho
Kyoto 619-02
Japan.

NAME

cepf0_stitch – stitch separate pitch and cepstra files

SYNOPSIS

cepf0_stitch **–p** *pitch file* **–c** *cepstra file* **–o** *pitcep file*

[**–n** *cepstrum order*]

[**–f** *sampling frequency*]

[**–b** *break number*]

DESCRIPTION

The pitch and corresponding cepstra information given in the two separate files [–p] *pitch file* and [–c] *cepstra file* respectively are stitch together into the single file [–o] *pitcep file*. The *pitch file* and *cepstra file* are assumed to contain binary data of type 'float'. Pitch values equal to [–b] *break number* are taken to represent unvoiced speech. All other pitch values represent a pitch frequency in Hertz. If a pitch value represents unvoiced speech, it is set to zero; otherwise if is converted to a pitch period in units of number of samples (the sampling rate given by [–f] *sampling frequency*). The cepstral coefficients are assumed to be given as a series of structures containing [–n] *cepstrum order* plus one (for the zeroth cepstral coefficient) values. The converted pitch values and corresponding cepstral coefficients are written to the *pitcep file* as a series of pitch value followed by cepstral coefficient values, structures. If the lengths of the two input files is incompatible, the stitch terminates when data runs out for the shorter.

OPTIONS

–p *pitch file* specified *pitch file* as the headerless, binary file containing pitch values in Hertz. The binary data must be of type 'float'.

–c *cepstra file* specifies *cepstra file* as the headerless, binary file containing a series of parallel structures of 'float' type cepstral coefficient values.

–o *pitcep file* specifies *pitcep file* as the file to be created with the pitch and cepstral coefficient values combined. This will be a headerless file containing data of type 'float'.

–n *cepstrum order* specifies the number of cepstral coefficients given in the cepstra file as *cepstrum order*, excluding the zeroth coefficient which must always be given. The default is 30 (plus one) cepstral coefficients.

–f *sampling frequency* specifies the sampling rate in which terms the pitch period is to be calculated, as *sampling frequency*, **in Hertz. Default is 12000Hz.**

–b *break number* **breaks in the F0 contour described in the file** *pitch file* **which represent unvoiced speech are given by the value** *break number*. **The default setting is 0.0.**

SEE ALSO

cep_fft (PCB) – fast Fourier transform based (improved) cepstral analysis.
cep_syn (PCB) – cepstral speech synthesis.
cepf0_split (PCB) – split joint pitch and cepstra file.

BUGS

None known.

AUTHOR

Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai
Seika-cho
Kyoto 619-02
Japan.

NAME

compf0 – compare files describing F0 contours in Xmg format

SYNOPSIS

compf0 *correct_F0_file pda_F0_file* [*appended_stats_file*]

DESCRIPTION

The two F0 contour description files, *correct_F0_file* and *pda_F0_file* are assumed to contain headers for Xmg and data in ascii format as generated by **pm_freq** and **adlb_xmg**. These files may be asynchronous representations of F0 contours and may be of different lengths. At times where the F0 value is known for one of the contours (from the file) but not the other, the unknown one is inferred from its data by interpolating between the two F0 points known to lie at either side. The files are thus internally synchronised. A contour is assumed to represent unvoiced speech for times outside the scope of its data description.

The comparison of the contours yields the following statistics:

The total duration in milliseconds, for which the correct F0 contour given in *correct_F0_file* represents unvoiced speech (breaks indicated by an equals character (=)), but the pda F0 contour given in *pda_F0_file* represents voiced speech (ie. Unvoiced errors).

The total number of continuous sections for which unvoiced errors occurred.

The total duration in milliseconds, for which the correct F0 contour represents voiced speech, but the pda F0 contour represents unvoiced speech (ie. Voiced errors).

The total number of continuous sections for which voiced errors occurred.

The number of synchronous comparisons in which the ratio, (*pda F0 value* – *correct F0 value*) divided by *correct F0 value* is greater than 0.2, and both contours represent voiced speech (gross error (pitch double or more?)).

The number of synchronous comparisons in which the ratio, (*correct F0 value* – *pda F0 value*) divided by *correct F0 value* is greater than 0.2, and both contours represent voiced speech (gross error (pitch half or less?)).

The average distance (Hz) between the contours at times when both represent voiced speech, excluding when gross errors occur.

The population standard deviation of the distance (Hz) between the contours at times when both represent voiced speech, excluding when gross errors occur.

These statistics are written to the file stream *stdout* in a user readable form, or appended to the end of the file *appended_stats_file*, if given, in an abstact data form.

OPTIONS

appended_stats_file write statistics of comparison to the end of file *appended_stats_file*. **The default is the file stream** *stdout*.

SEE ALSO

adlb_xmg (PCB) – convert an Audlab headed track and label files to Xmg format.
mix_stats (PCB) – accumulate F0 contour comparison statistics.
pm_freq (PCB) – pitch mark data to Xmg–format F0 contour.

DIAGNOSTICS

The diagnostics produced by COMPF0 are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

diac2label – form label files from the diacritics of a label file

SYNOPSIS

diac2label –f *configuration_file* –i *label_file*
[–s *"silence_label"*] [–p]

DESCRIPTION

The diacritics of the [–i] *label_file* are separated into new label/diacritic files, as specified in the [–f] *configuration_file*. The format of the configuration file is a sequence of three line "file/label/diacritic" descriptors:

SET:*ext*:
lebal
citircaid

where *ext* is a three character file extension appended to the root name of the input *label_file* to form the name of the new output label file. The lines *lebal* and *citircaid* must contain single strings which specify those diacritics in [–i] *label_file* which form the labels and diacritics respectively in the new output file.

If the [–p] flag is passed, the output label files are padded with silence labels specified by [–s] *"silence_label"* to the start and the end of the input label file and at any intermediate points where an output label is not specified by the diacritics of the input file.

The label "#" is reserved and should not be included in any of the *lebal* fields of the configuration file. This character is automatically appended to the diacritic list of all input labels which are identical to *"silence_label"*. Occurrences of "#" are then modified to *"silence_label"* when output. This has the effect of making the onset of each silence label in the input file into a syllable boundary.

OPTIONS

–f *configuration_file*	gives the name of the file specifying the sets of new labels and their diacritics, and requested file name extensions.
–i *label_file*	gives the file name of the label file whose diacritics at to be extracted and whose root name is used in generating the file names for the new label files.
–s *"silence_label"*	specifies the silence label string to be used during padding. The default string is "##".
–p	use *silence_label* padding.

DIAGNOSTICS

The diagnostics produced by DIAC2LABEL will inform the user of any misuse at the command line, and of any file i/o or memory allocation errors.

BUGS

The program does not check that the file extensions for the new label files specified in the configuration file are unique and that those specified differ from the extension of the input file. Care must therefore be taken not to attempt to overwrite the input file!

Although more than two diacritic/label sets can be defined in the configuration file, the program has been customised for only two sets. The first of these sets is used to define the boundaries of the second (or subsequent) sets. A future enhancement to the program should allow the configuration file to specify which set defines the boundaries of any other set.

Reservation of the character "#" should be removed, an alternative method should be used to make occurrences of the silence label in the input file into syllable boundary markers. In fact, this should be modified so that the configuration file specifies for which output files this behaviour should be applied, and which input labels should activate it.

AUTHORS

Keith Edwards and Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

dur – determine total unvoiced and voiced durations from Xmg F0 contours.

SYNOPSIS

dur *F0_file* ...

DESCRIPTION

One or more *F0_files* in Xmg format are read. The program accumulates the total durations for which F0 is defined (representative of voiced speech) and the inter–voiced regions (assumed to be unvoiced). The utterance is taken to be silent at times preceding the first voiced section and at times after the last voiced section.

OPTIONS

None.

SEE ALSO

mix_stats (PCB) – accumulate F0 contour comparison statistics.
pm_freq (PCB) – pitch mark data to Xmg–format F0 contour
pmlar (PCB) – a pitch marker for laryngograph data

DOCUMENTATION

XMG, a multiple graph drawer for X11.

DIAGNOSTICS

The diagnostics produced by DUR are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

energy – calculate normalised energies within specified frequency bands.

SYNOPSIS

energy –i *audlab_vox_file* –o *energy_file*
[–p *power of 2 transform size*]
[–w *frame length*]
[–s *frame shift*]
[–f *lower_cut–off_frequency upper_cut–off_frequency*] ...

DESCRIPTION

Segments of data of length [–w] *frame length* (ms) are taken from the speech data in [–i] *audlab vox file*. Successive frames are shifted by [–s] *frame shift* (ms). Each frame is passed through a Blackman–Harris window. The amplitude spectrum |F(nf)| is then calculated for each windowed frame by a Fast Fourier Transform using (2 to the power of [–p] *power of 2 transform size*)-points. The FFT size must be big enough to contain all the samples in a frame. Zero padding is used to fill excess points. The total energy (summation of frequency bin amplitudes) is determined for each range of frequencies specified by the [–f] flag. For each frequency band, the maximum frame energy in the utterance is found. The frequency band energies in each frame are then expressed in decibels with respect to their corresponding maximum energy ie. 20 log10 |F(nf)/Fmax(nf)|.

OPTIONS

–i *audlab_vox_file* specifies *audlab vox file* as input to the algorithm. Must be an AUDLAB file containing sample data of type 'short'.

–o *energy_file* specifies *energy file* as output from the algorithm. This will be an AUDLAB format file containing a number of interleaved (parallel) tracks or a single (serial) track; depending upon the number of frequency bands for which the energy is calculated. The track descriptors are given names, eg. "e0-2000", to describe the frequency band (0Hz to 2000Hz) for which the energy is calculated.

–p *power of 2 transform size* set the number of points to be used in the FFT to two to the power of *power of 2 transform size*. Must be greater than five. Default setting is 9 (a 512–point FFT).

–w *frame length* set the length of the analysis frame to *frame length* in milliseconds. Must be greater than zero. Default frame length is 25.6ms.

–s *frame shift* set the shift between consecutive analysis frames to *frame shift* in milliseconds. Must be greater than zero. Default frame shift is 6.4ms.

–f *lower_cut–off_frequency upper_cut–off_frequency* defines the range of FFT frequency bins (Hz) whose amplitudes are accumulated in calculating energy. The *lower_cut–off_frequency* must be a positive value. The *upper_cut–off_frequency* must be greater than *lower_cut–off_frequency* and less than half the data sampling frequency. Any number of ranges may be specified. The output file will contain a track for each frequency band. The default is one range from 0.0Hz to 2000.0Hz.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

DIAGNOSTICS

The diagnostics produced by ENERGY are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

f0_hist – histogram, mean and standard deviation of Xmg F0 contours.

SYNOPSIS

f0_hist *F0_file* ...

DESCRIPTION

Assumes a sampling frequency of 20kHz. 100 histogram bins exist, each with a band of 5Hz. (These values shouldn't be fixed.) The series of *F0_file*'s (Xmg format, F0 segment files) are read. Each F0 value results in the corresponding histogram bin being incremented. An error occurs if the input frequency is greater than 500Hz. (bug! well sort of). The program returns the counts in each histogram bin, the mean and population standard deviation of the F0 values, and the mean and population standard deviation of the error in F0 values that may be resultant of time quantisation errors during sampling. These statistics are probably the most useful part of the program.

OPTIONS

None.

DOCUMENTATION

XMG, a multiple graph drawer for X11.

DIAGNOSTICS

The diagnostics produced by F0_HIST are intended to be self–explanatory.

BUGS

Loads of them! It's a terrible program and should never have been written. Use A.P.S. if possible.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

gcipd – a pitch determination algorithm based on glottal closure instants

SYNOPSIS

gcipd –i *audlab sample file* –o *pitch file*

[–L *lpc order*]

[–w *frame length*]

[–s *frame shift*]

[–l *lower pitch frequency limit*]

[–u *upper pitch frequency limit*]

DESCRIPTION

Frames of low–pass filtered data are read from the input file, [–i] *audlab sample file* and analysed in turn. The frame length is given by the [–w] flag and the distance between the beginnings of consecutive frames is given by the [–s] flag.

A number of linear prediction coefficients, dictated by the [–L] flag, are calculated for the frame being analysed. This LPC analysis comes from Witten (1982). The coefficients are used to produce a wavelet c(n) due to an epoch.

$$c(n) = \sum_{i=1}^{P} a_i.s(n-i) \tag{1}$$

Only 80 samples of this signal are determined. c(n) is then cross–correlated with the original frame of samples to form a maximum–likelihood epoch determination (MLED) signal.

$$f(n_0) = \sum_{n=0}^{N-1} s(n+n_0).c(n) \quad n_0 \geq 0 \tag{2}$$

A glottal closure instant determination signal is calculated from the MLED signal by performing average subtraction. The estimates of F0 are made from the distances between two consecutive, prominent peaks in the GCIDS. The prominent peaks are found using an approach based on Reddy (1967).

The fundamental frequency (and period) for each frame is written to the output file given by the [–o] flag, in Audlab format.

OPTIONS

–i *audlab sample file* specifies *audlab sample file* as input to the algorithm. The file must be an AUDLAB headed file containing low–pass filtered sample data of type 'short'.

–o *pitch file* specifies *pitch file* as output from the algorithm. Will be an AUDLAB file containing two interleaved tracks. The first track, with a track descriptor of "gcipd_f0", gives the fundamental frequency (Hz) and the second, with the descriptor "gcipd_per", gives the pitch period (ms).

–L *lpc order* set the number of LPC coefficients to be determined during analysis to *lpc order*. The default value is twelve. The maximum order that can be set is thirty.

–w *frame length* set the length of the analysis frame to *frame length* in milliseconds. Must be greater than zero. The default id 25.6 ms.

–s *frame shift* set the shift between consecutive analysis frames to *frame shift* in milliseconds. Must be greater than zero. The default is 5.6 ms.

–l *lower pitch frequency limit* set the lowest pitch frequency allowed to *lower pitch frequency limit*. Must be greater than zero. The default is 40.0 Hz.

–u *upper pitch frequency limit* set the highest pitch frequency allowed to *upper pitch frequency limit*. Must be greater than current *lower pitch frequency limit*. The default is 400.0 Hz.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

Y.M. Cheng, D. O'Shaughnessy, "Automatic and reliable estimation of glottal closure instant and period," IEEE Trans. Acoustics, Speech, and Signal Processing ASSP-37 No.12 pp.1805-1814 (1989)

I.H. Witten, 'Principles of Computer Speech,' Academic Press Inc., London (1982)

D.R. Reddy, "Pitch period determination of speech sounds," Communications of the Association of Computing Machinery Vol.10 No.6 pp.343-348 (1967)

DIAGNOSTICS

The diagnostics produced by the GCIPD algorithm are intended to be self-explanatory.

BUGS

None Known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

getminload – get name of host with minimum load.

SYNOPSIS

getminload [*filename*]

DESCRIPTION

The file *filename* (or the file named **rupout**, by default) is examined. This file must contain the standard output generated by the UNIX program *rup*. The program quantises the long term average loads to the nearest half unit and then returns the name of the host with the minimum quantised load. If more than one machine corresponds to the minimum load, the first machine name in the file list is returned.

OPTIONS

filename name of the file containing the standard output generated by *rup*. Default name is **rupout**.

EXAMPLE

In order to determine which of the six hosts named liddell, scott, brodie, minto, phoenix and watt currently has the minimum load quantised to the nearest half unit with preference to liddell, then to scott, etc., use:

```
% setenv MINHOST 'rup liddell scott brodie minto phoenix watt > /tmp/rupout;
  getminload /tmp/rupout; rm -f /tmp/rupout'
```

The environment variable MINHOST will contain the name of the host with the minimum load.

SEE ALSO

rup (1C) – show host status of local machines (RPC version).

DIAGNOSTICS

Returns -1 if *filename* (or **rupout**) cannot be opened.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

getpid – get process ID for a given name from a file.

SYNOPSIS

getpid *mode signal* [*filename*]

DESCRIPTION

Reads **stdin** to get the name of a program. The file *filename* (or the file named **psout**, by default) is examined for occurrences of the program name. The file should be of the format output to **stdout** by *ps*. If *mode* is set to zero, all processes beginning with the name given are selected. If *mode* is equal to one, only those processes whose name matches exactly are selected. For each matching process, the corresponding 'PID' is found and the message,

kill -*signal* 'PID'

is printed to **stdout**.

OPTIONS

filename name of the file containing the standard output generated by *ps*. Default name is **psout**.

SEE ALSO

ps (1) – display the status of current processes.

DIAGNOSTICS

Returns -1 if *filename* (or **psout**) cannot be opened.

BUGS

None known.

AUTHORS

Paul Bagshaw and Fergus McInnes
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

hps – a pitch determination algorithm based on the harmonic product spectrum

SYNOPSIS

hps –i *audlab vox file* –o *pitch file*

[–F *power spectrum file*]

[–H *harmonic histogram file*]

[–p *power of 2 transform size*]

[–w *frame length*]

[–s *frame shift*]

[–l *lower frequency limit*]

[–u *upper frequency limit*]

[–t *v/uv threshold*]

[–k *compression factor*]

DESCRIPTION

The fundamental frequency of a periodic signal can be determined by measuring the frequencies of its higher harmonic components and computing the greatest common divider of these frequencies (Schroeder, 1968). For each harmonic frequency, an entry is made to a frequency histogram at the frequency of the harmonic and at integer divisions of the harmonic frequency. As a higher signal amplitude usually means a better accuracy and reliability of frequency measurement, each entry to the histogram can be weighted by a monotonicly increasing function of the harmonic amplitude or the logarithm of its amplitude. The frequency at the peak of the histogram represents the greatest common divider of the harmonic frequencies, and hence the fundamental frequency. The histogram can be determined from a short time amplitude spectrum of a segment of speech |F(nf)| (Noll, 1970) and represented as the 'harmonic sum spectrum',

$$S(f) = \sum_{n=1}^{N} \mid F(nf) \mid \tag{1}$$

Alternatively, the histogram can be formed from a short time log power spectrum 20 log10 |F(nf)| and represented as the 'harmonic product spectrum',

$$\log_{10} P(f) = \sum n = 1^{N} 20.\log_{10} \mid F(nf) \mid \tag{2}$$

hence,

$$P(f) = \prod_{n=1}^{N} \mid F(nf) \mid \quad (+10^{20}\, which\, is\, ignored) \tag{3}$$

The low-quefrency structure of the log power spectrum of speech only adds confusion when compressed to form the harmonic product spectrum, so they are removed by smoothing the spectrum with a window function W(f), (the 4-term Blackman-Harris window (Harris, 1978) in this case,) and then subtracting the smoothed spectrum from the original spectrum before calculating the product spectrum. The low-quefrency lifted log spectrum is,

$$L(f) = 20.\log_{10} \mid F(f) \mid - W(f).20\log_{10} \mid F(f) \mid \tag{4}$$

and the harmonic product spectrum,

$$P(f) = \prod_{n=1}^{N} L(nf) \tag{5}$$

N is given by [–k] *compression factor*. The fundamental frequency is given by the position of the maxima of P(f). If this maxima has an amplitude that is greater than [–t] *v/uv threshold*, the segment of speech being analysed is assumed to be voiced. If voiced, the pitch frequency estimate is ensured to be greater than or equal to [–l] *lower frequency limit* and less than or equal to [–u] *upper frequency limit*.

Segments of data of length [–w] *frame length* (ms) are taken from the speech data in [–i] *audlab vox file*. Successive frames are separated by [–s] *frame shift* in milliseconds. 20 log10 |F(nf)| is calculated for each frame by a Fast Fourier Transform using (2 to the power of [–p] *power of 2 transform size*)-points. The FFT size must be big enough to contain all the samples in a frame. Zero padding is used to fill excess points. The power spectrum of each frame may be passed to [–F] *power spectrum file*. P(f) is calculated from this for each frame. This harmonic power spectrum may be passed to [–H] *harmonic histogram file* for each frame analysed. The pitch frequency and pitch period are output in two tracks in the file [–o] *pitch file*, for each voiced segment. Zero is passed for all unvoiced and silent segments.

OPTIONS

–i *audlab vox file* specifies *audlab vox file* as input to the algorithm. Must be an AUDLAB file containing sample data of type 'short'.

–o *pitch file* specifies *pitch file* as output from the algorithm. Will be an AUDLAB file containing two interleaved tracks. The first track gives the fundamental frequency (Hz) and the second gives the pitch period (ms).

–F *power spectrum file power spectrum file* will contain the spectrum (dB) obtained by an FFT for each segment analysed, in the form of an AUDLAB spatial domain track.

–H *harmonic histogram file harmonic histogram file* will contain the harmonic product spectrum for each segment analysed, in the form of an AUDLAB spatial domain track.

–p *power of 2 transform size* set the number of points to be used in the FFT to two to the power of *power of 2 transform size*. Must be greater than six. Default is 12.

–w *frame length* set the length of the analysis frame to *frame length* in milliseconds. Must be greater than zero. Default is 25.6 ms.

–s *frame shift* set the shift between consecutive analysis frames to *frame shift* in milliseconds. Must be greater than zero. Default is 6.4 ms.

–l *lower frequency limit* set the lowest pitch frequency allowed to *lower frequency limit*. Must be greater than zero. Default is 40 Hz.

–u *upper frequency limit* set the highest pitch frequency allowed to *upper frequency limit*. Must be greater than current *lower frequency limit*. Default is 500 Hz.

–t *v/uv threshold* set the 'maximum bin amplitude in harmonic product spectrum' threshold between voiced and unvoiced speech to *v/uv threshold*. Must be greater than zero. Default is 2.0e9.

–k *compression factor* set the number of frequency bins summed in formation of the harmonic product spectrum to *compression factor*. Must be greater than one. Default is 5.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

F.J. Harris, "On the use of windows for harmonic analysis with discrete Fourier transform," Proc. IEEE Vol.66 No.1 pp.51-83 (1978).

A.M. Noll, "Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate," in 'Symposium on Computer Processing in Communication,' Polytechnic Institute of Brooklyn Microwave Research Institute, New York, Vol.19, pp.779-797 (1970).

M.R. Schroeder, "Period histogram and product spectrum: new methods for fundamental-frequency measurement," Journal of the Acoustical Society of America Vol.43 No.4 pp.829-834 (1968).

DIAGNOSTICS

The diagnostics produced by the HPS algorithm are intended to be self-explanatory.

BUGS

None Known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

labelformat – convert between Segstat, HMM, Audlab, and Waves label file formats.

SYNOPSIS

labelformat **–i** *label file* **–o** *new label file*

[**–f** *output label format*]

[**–s** *sampling frequency*]

[**–d** *diacritic symbols file*]

[**–c** *separator character*]

[**–n**]

DESCRIPTION

The speech segmentation file [–i] *label file* is identified as being in Segstat, HMM, Audlab, or Waves format, automatically. It is converted to a new format given by [–f] *output label format* (either **segstat**, **hmm**, **audlab** or **waves**) and placed into the output file [–o] *new label file*. If the input or output file is in Segstat format, the sampling frequency given by the [–s] flag is used. If the input or output file is in Audlab format, the diacritic symbols file given by the [–d] flag is used. The format of the input file and the format of the output file must differ. If they are the same, a message indicating that no conversion is required, is returned.

The segmentation file formats recognised are:

segstat
The file contains a list of segmentations, one per line, with each end of line indicated by the character '\n'. Each segmentation is described by a start time in samples, stop time in samples, label character string, and an optional diacritic characters string. Each field is separated by a white space character. The start and stop times are read as integers.

start_time stop_time label [*diacritics*]

The start time of the first segmentation in the file should be zero (samples). Start and stop times in samples are converted to and from times in milliseconds and seconds by using the [–s] *sampling frequency*.

hmm
The file contains a list of segmentations, one per line, with each end of line indicated by the character '\n'. Each segmentation begins with an open bracket '(' character and is described by a start time in milliseconds, stop time in milliseconds, label character string, and an optional diacritic characters string. The segmentation ends with a close bracket ')' character. Each field is separated by a white space character. The start and stop times are read as floats.

(start_time stop_time label [*diacritics*] *)*

The start time of the first segmentation in the file should be zero (ms).

audlab
Audlab segmentation files begin with two headers, a general file header and a label file header. Each segmentation is described by a binary structure which consists of two floats giving the start time and stop time in seconds, a label string, thirty two characters long, and a 32-bit diacritics number.

general file header
label file header
segmentation data structure ...

Each bit set in the diacritics field is related to a diacritic character given by the [–d] *diacritic symbols file* to form and encode a diacritics character string. The default file headers of the output file give a file start time of 0.0s, a file stop time corresponding to the stop time of the final segment listed, the file description as "labelformat", the label method as HAND, and the label type as PHONEMIC.

waves
Waves segmentation files input to this conversion program must have been generated using **xlabel**(1–ESPS). These files consist of a header section (which may be empty) of keyword-value pairs, and a segmentation information section containing one segment description per line. The sections are separated by a hash character (#) occupying a single line.

signal speech
type 0
color 121
comment created by labelformat
font -adobe-helvetica-bold-r-*-*-12*
separator ;
nfields 2
#
end_time colour_code label
end_time colour_code label;diacritics

The *signal*, *type*, *color*, *comment* and *font* fields of the header section are not used. Each segmentation record has three entries separated by white spaces. These are, a label end time in seconds (read as type float), a colour code (not used), and a multi–field character string. The first label is assumed to start at 0.0s and labels are assumed to be consecutive and in chronological order (this is checked by the program). The multi–field character string contains a maximum number of fields given by the *nfields* keyword in the header (default one), and are separated by the character given by the *separator* keyword (default semicolon). The first field describes the segment label and the second (optional) field describes any diacritics. Any additional fields are ignored during conversion. The separator used in an output file of this format is specified by [–c] *separator character.*

The diacritics field given in the input file can be completely ignored by passing the [–n] flag to form an output file with no diacritic information.

OPTIONS

–f *output label format*	used to inform the program of the label format the output file *new label file* is to be in. The *output label format* must be specified as segstat, hmm, audlab, or waves. The default output format is audlab.
–s *sampling frequency*	specifies the sampling frequency used for data represented in the Segstat label file format as *sampling frequency* in Hz. This must be a non-zero, positive integer. Default is 20000Hz.
–d *diacritic symbols file*	specifies the location (full path name) and name of the Audlab format, diacritic symbols description file. The default is $HOME/defaults/diacrit.sym.
–c *separator character*	the character used for output files in the Waves label format to separate the label string and the diacritics information is specified by *separator character*. The default is a semicolon (;).
–n	no diacritic information is written to the output file when this flag is passed.

SEE ALSO

xlabel (1–ESPS) – time series labelling attachment for xwaves+

DIAGNOSTICS

The diagnostics produced by LABELFORMAT are intended to be self–explanatory.

BUGS

None that I know of, yet.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

mix_stats – accumulate F0 contour comparison statistics.

SYNOPSIS

mix_stats *stats_file uv_dur v_dur*

DESCRIPTION

The program reads in the statistics file *stats_file* that must have been produced by **compf0** and accumulates the values therein. The totals are then output to the file steam *stdout*. The duration of unvoiced and voiced errors are expressed as a percentage of the total duration of voiced speech (*v_dur*) and the total duration of unvoiced speech (*uv_dur*), determined by **dur**, respectively.

OPTIONS

None.

SEE ALSO

compf0 (PCB) – compare files describing F0 contours in Xmg format
dur (PCB) – determine total unvoiced and voiced durations from Xmg F0 contours.

DIAGNOSTICS

The diagnostics produced by MIX_STATS are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

normf0 – declination compensation and normalisation of F0 contours

SYNOPSIS

normf0 –i *infile* **–o** *outfile*

[**–t** *F0 track number*]

[**–d** [**–c**] [**–p** *permitted microprosodic perturbation*] [**–w**] [**–l**]]

[**–n** [**–m** *mean F0*] [**–s** *population standard deviation of F0*]]

DESCRIPTION

The algorithm implemented attempts to compensate for declination in fundamental frequency and normalise differences of pitch level and range between speakers.

Fundamental frequency (F0) values are read in from the [–t] *F0 track number*'th track of the AUDLAB headed file [–i] *infile*. If the [–d] flag is passed, declination compensation is applied. If the [–n] flag is passed, fundamental frequency normalisation is performed. When both are passed, declination compensation is applied first (with the mean and the population standard deviation of the contour retained by frequency shifting and scaling) and then the normalisation is performed. The resultant modified fundamental frequency contour is written to the AUDLAB headed file [–o] *outfile*.

Declination Compensation (Huber, 1989).

Linear interpolation is used across intervals of aperiodic speech and silence (for which F0 is undefined and the F0 contour is described by breaks) that occur between two segments of periodic speech, to give a continuous contour. The starting point and finishing point of the continuous contour, and all turning points along this contour are made candidates for peaks and valleys. A candidate is added to a list of peaks {valleys} if the F0 value for the candidate is both greater {less} than all F0 values since the previous valley {peak} or beginning point (except in the case of the starting point as a candidate), and greater {less} than all F0 values up to the next valley {peak} or end point (except in the case of the finishing point as a candidate). A further constraint is imposed to take into account the effect of microprosodic perturbation, in that adjacent peak and valley candidates must have F0 values which differ by at least [–p] *permitted microprosodic perturbation*. Flat–top peak points and flat–bottom valley points are allocated a turning point time which lies in the centre of the flat region. Such flat turning points may be a consequence of quantisation errors occurring in F0 determination algorithms. All other significant points are allocated a turning point time which relates to the instance the peak or valley occurs. If the [–w] flag is passed, the significant peaks and valleys (ie. those candidates that satisfy the above criteria) are listed in an AUDLAB track labelled "signif_pnts".

The lists of significant peaks and significant valleys are analysed separately so as to find a topline and a baseline respectively. If the [–l] flag is passed, this is done by using the robust linear regrssion method based on least median of squared residuals (Rousseeuw, 1984); otherwise the following process is used which incorporates least squares linear regression (Press, et. al., section 14.2, 1986). Initially, least squares linear regression is used to

determine the straight line which best fits all the points in a list of significant peaks or valleys. Pearson's linear correlation coefficient (Press, et. al., section 13.7, 1986) is then used to quantify the degree of "fit" between the F0 value as predicted by the straight line at each turning point time, and that as given in the list. If there are more than three significant points in a list, further analysis is performed to find the greatest number of successive points in a list whose linear regression line has a better fit (ie. higher Pearson's coefficient with those points selected) than using all the points. This further analysis aims to find the optimal set of points in a list to be used in establishing the topline or baseline. The analysis starts be using the total number of points less one, and decrements the number of points used by one, until only three points are being consider. For a number of points less than three, linear regression results in a perfect fit, with a Pearson's coefficient of one. This gradual reduction in the number of points considered is terminated when a set of points are found to constitute a better fit than using all of the points. Every possible combination of successive points (however many there may be) in the list is considered. For example, if there are a total of five points in a list and three successive points are being considered, the first three of five will be analysed in addition to the centre three of five, and the final three of five. The combination with the highest Pearson's coefficient is the one used for a given number of points. The straight line through the optimal points makes up the topline {baseline} for the list of peaks {valleys}.

Once a topline and a baseline have been found, they are vetted in the following manner. If there was only one significant peak {valley} found in the F0 contour, then the topline {baseline} is invalid. Furthermore, any line with positive slope is not representative of declination and is also branded invalid. If both lines are invalid, no declination can be detected in the contour and it is left unaltered. If only one line remains valid, then it is taken to represent the overall declination. If both lines are valid, but the [–c] flag is passed or their point of intersection lies within the duration of the F0 contour, then the average line (between the topline and baseline) is used to describe the overall declination. Otherwise, two gradually converging declination lines have been found with a pitch span varying with time. The F0 contour is shifted so as to level the declination line(s) to a horizontal position. When pitch span has be found, this process also involves some scaling of the contour. The contour is then adjusted further in order to retain its initial mean and population standard deviation.

If two converging declination lines where used in the above, the topline and baseline are placed in [–o] *outfile* as AUDLAB tracks labelled "lin_regr_pks" and "lin_regr_vlys" respectively. If no valid lines were found, the tracks are set to zero; otherwise the declination line used is placed in the track labelled "lin_regr_pks" regardless of whether it was in fact the topline, baseline or average line that was used.

Normalisation (Rose, 1987).

The Z–score normalisation scheme is implemented, for which the normalised value of F0 is set to (*input F0 – long term mean F0*) divided by *long term popular standard deviation.* Thus the normalised F0 contour is centred around the zero with each unit representing a frequency one population standard deviation from the mean. The *long term mean F0* may be set using [–m] *mean F0.* The *long term population standard deviation* may be set using [–s] *population standard deviation.*

The final modified F0 contour is written to the file [–o] *outfile* in an AUDLAB track labelled "normf0".

OPTIONS

–i *infile* specifies *infile* as input to the algorithm. Must be an AUDLAB file containing track data of type 'float'.

–o *outfile* specifies *outfile* as output from the algorithm. Will be an AUDLAB file containing a single track of modified fundamental frequency values and any debug tracks when the [–w] flag is also passed.

–t *F0 track number* specifies that the *F0 track number*'th track in [–i] *infile* contains the fundamental frequency values to be modified by the algorithm. The default value is 1.

–d compensate for declination if possible.

–c use average declination line whenever possible, rather than adjusting for pitch span when two declination lines (one topline and the other baseline) are found.

–p *permitted mircoprosodic perturbation* set the permitted level of variation in fundamental frequency that is not to be treated as constituting to a significant turning point when determining the declination to *permitted microprosodic perturbation.* The default is 5 Hz.

–w make debug declination tracks in [–o] *outfile*, namely the significant turning points found in determining the declination, the linear regression line for the peaks, and the linear regression line for the valleys.

–l use least of median squared residuals linear regression in determining the topline and baseline, rather than the method involving least squares linear regression and the Pearson's linear correlation coefficient.

–n normalise absolute values based on long term mean and population standard deviation.

–m *mean F0* set the long term mean fundamental frequency value to *mean F0* (Hz). The default is 130.0 Hz.

–s *population standard deviation* set the long term population standard deviation for the fundamental frequency to *population standard deviation* (Hz). The default is 35.0 Hz.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

D. Huber, "A statistical approach to the segmentation and broad classification of continuous speech into phrase–sized information units," Proc. IEEE ICASSP–89, Glasgow, Vol.1, pp.600–603 (1989).

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, 'Numerical Recipes: The Art of Scientific Computing,' Cambridge University Press, Cambridge, U.K. (1986).

P.J. Rousseeuw, "Least median of squares regression," Journal of the American Statistical Association Vol.79 pp.871–880 (1984).

P. Rose, "Considerations in the normalisation of the fundamental frequency of linguistic tone," Speech Communication Vol.6 No.4 pp.343–352 (1987).

DIAGNOSTICS

The diagnostics produced by the NORMF0 algorithm are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

pc2adlb_vox – convert PC speech files to Audlab headed format.

SYNOPSIS

pc2adlb_vox **–i** *no–header file* **–o** *audlab vox file*

[**–h**] [**–p**] [**–s**]

[**–f** *sampling frequency*]

[**–c** *anti–aliasing cutoff frequency*]

[**–b** *ADC resolution*]

[**–B** *output number of bits*]

[**–n** *number of parallel input channels*]

[**–u** *channel to use*]

DESCRIPTION

An Audlab header is attached to the speech sample data in *no–header file* unless the [–h] flag is passed. Each data sample must occupy two bytes. The program sets the header fields as follows:

struct fileheader

The segment ID name is set to the root basename of the input file [–i] *no–header file*. The segment ID start time is set to 0.0s and the stop time is set in accordance with the sampling frequency and number of samples taken in each channel of input data. The file header description string is "Database". All other fields are intrinsic to the type and size of [–o] *audlab vox file* ie. sample data of type short.

struct sampleheader

The number of channels contained in the output [–o] *audlab vox file* is set by [–n] *number of parallel input channels*; unless the [–u] *channel to use* is passed, in which case only one channel is output. For one channel of data, the parallel/serial flag is set to SERIAL; otherwise, channels are stored in PARALLEL format. The sampling frequency is set to 20,000Hz unless explicitly changed by passing [–f] *sampling frequency*. Speaker reference is "name", age 24, and male. The microphone details are set to "Shure SM10A" and the ambient conditions as "CSTR Rec. studio".

struct sampledescriptor

If only one channel of data exists, the channel descriptor is "close-talking"; otherwise "channel n", where n is the channel number (starting with 1). The channel length corresponds exactly with the number of data samples that can be read for each channel. The number of bits in the output data is specified by [–B] *output number of bits*. The default number of output bits is inherited from [–b] *ADC resolution*, which in turn has a default value of 16 bits. If the number of output bits and the resolution of the

ADC differ, the sample data is scaled by multiply or dividing by two until the desired bit size is obtained. The anti–aliasing filter cut–off frequency field is set to [–c] *anti–aliasing cutoff frequency* (usually half the sampling frequency or less). It is assumed to be the same for all channels. The default Audlab display values set in the structure are designed to allow the entire speech data to be viewed.

The sample data is transferred from [–i] *no–header* to [–o] *audlab vox file.* Initially, byte swapping is performed, uf the [–s] flag is passed. The polarity of the data my be inverted by multiply each sample by minus one, by passing the [–p] flag. Finally, any scaling required to adjust the number of bits occupied by the data is done.

OPTIONS

–i *no–header file* a headerless file of speech data named *no–header file* is passed as the input. The number of samples contained in the file must be an exact multiple of the number of channels recorded. Multi–channel data is assumed to be in a parallel format.

–o *audlab vox file* the Audlab headed (optional) output file created will be named *audlab vox file.* All necessary headers and byte manipulation are included.

–h prevent an Audlab header from being attached.

–p reverse the data polarity, ie. multiply all values by -1.

–s byte swap. Useful when converting DOS data to SUN data.

–f *sampling frequency* specify data sampling frequency in Hertz. Default value is 20000Hz.

–c *anti–aliasing cutoff frequency* specify anti–aliasing filter cut–off frequency in Hertz. Default setting is 10000.0Hz.

–b *ADC resolution* set the analogue–to–digital converter resolution (in bits) used to form the data in [–i] *no–header file.* The default is 16 bits.

–B *output number of bits* set the number of bits the output data in [–o] *audlab vox file* is to take. The default is the same as *ADC resolution.* The input data is scaled my multiplication or division by orders of 2 in order to adjust the number of bits used by the data.

–n *number of parallel input channels* specifies the number of channels present in the input file. Input samples of multi–channel data are assumed to be in a parallel format. All channels will be written to the output file unless one specified channel is specified.

–u *channel to use* select only one channel from a multi–channel input file.

SEE ALSO

alter_adlb_hd (PCB) – alter Audlab file headers.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

DIAGNOSTICS

The diagnostics produced by PC2ADLB_VOX are intended to be self-explanatory.

BUGS

If the output number of bits is set to a size greater than sizeof (short), data might be lost.

Some of the header fields are "hard wired". Additional options should be included to allow them to be altered. The use of an optional defaults file is not recommended as the format of such a file would require a standard (which limits the software maintenance) and most of the fields must be set explicitly by the program or already permit alternative values to be selected at the command line. Fields that should also be allowed to change are: fileheader.descr, sampleheader.speaker_id, sampleheader.age, sampleheader.sex, sampleheader.mike, sampleheader.ambience and sampledescriptor.descr (see Audlab documentation).

AUTHOR

Paul Bagshaw and Eddie Rooney
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

piecewise – an F0 contour piecewise stylisation algorithm

SYNOPSIS

piecewise **–i** *infile* **–o** *outfile*
[**–t1** *F0 track number*]
[**–t2** *weight track number*]
[**–l** *initial window length*]
[**–v** *permitted F0 variation*]
[**–d** *minimum discontinuity duration*]
[**–h**] [**–r**]

DESCRIPTION

The algorithm implemented forms a piece–wise stylised F0 contour in the manner described in (Bagshaw, 1992), with optional weighting, semitone to Hertz conversion, and type of linear analysis. The stylisation aims to eliminate segmental variations in an F0 contour while retaining suprasegmental trends.

Fundamental frequency values are read in from the [–t1] *F0 track number*'th track of the Audlab type headed file [–i] *infile*. If [–t2] *weight track* is non-zero, the F0 reliability measure, such as that given by **srpd**(PCB), is read from the specified track in the features file [–i] *infile*. The F0 contour is ideally pre-smoothed using the non–linear smoothing algorithm implemented in the program **smoother**(PCB).

Turning–points in the F0 contour are located by applying linear regression analysis to a window of voiced frames whose length is initially set to [–l] *initial window length*. The type of linear regression analysis used is the robust least median of squared residuals regression (l.m.s.r.) unless the [–r] flag is passed, in which case least squares (l.s.) linear regression is used. If the track containing weights has not be set to zero, then only those F0 values whose reliability measure is greater than 0.9 are used in the l.m.s.r. analysis; otherwise, or if l.s. analysis is requested, all values are used. Turning–points occur only at points when the absolute difference between the actual F0 and predicted (from the regression coefficients) F0 values is greater than [–v] *permitted F0 variation*, and this situation arises for all following frames up to either the final voiced frame in the contour or such that the duration of this discontinuity is greater than [–d] *minimum discontinuity duration* (which ever occurs first).

The points found are modified to prevent contour discontinuities other than at the boundaries between unvoiced and voiced speech. The new stylised contour is created by linear interpolation of F0 between them and reseting each frame that was unvoiced in the non-stylised contour to an unvoiced state in the new one. The resultant data is coverted back to a Hertz scale if the [–h] flag is passed, otherwise it remains in a semitone scale; and it is written to the output file specified by [–o] *outfile*.

A second contour of F0 residuals is also written to the [–o] *outfile*. This contour is derived

by subtracting each F0 value in the new stylised contour from the original F0 contour and adding the mean fundamental frequency value of the original F0 contour. The residuals contour would ideally contain only the segmental variations of pitch.

OPTIONS

–i *infile* specifies *infile* as input to the algorithm. Must be an Audlab file containing track data of type 'float'. This input file is required to contain a track describing the fundamental frequency of a speech waveform in Hertz and an optional track giving a reliability measure for each F0 value.

–o *outfile* specifies *outfile* as output from the algorithm. This will be an Audlab file containing two tracks of type 'float' in serial format. The first of these tracks is the piece–wise stylised fundamental frequency contour and the second is the F0 residuals contour.

–t1 *F0 track number* specifies that the *F0 track number*'th track in [–i] *infile* contains the fundamental frequency values to be modified by the algorithm. The default value is track 1 (the first track).

–t2 *weight track number* specifies that the *weight track number*'th track in [–i] *infile* contains the reliability measure for each F0 value. The default value is track 2 (the second track). If this option is set to zero, then no weight track is believe to be available.

–l *initial window length* specifies the length of the initial linear regression window. The default value is 5 frames.

–v *permitted F0 variation* specifies the maximum variation in F0 that is permitted without the deviating F0 value being considered as an irregularity of the F0 contour or as part of a new piece–wise linear section of the stylised contour. The default permitted variation is one semitone.

–d *minimum discontinuity duration* sets the minimum duration a discontinuity in the F0 contour may occur without constituting a new piece–wise section of the stylised contour. The default is 100 milliseconds.

–h output values in Hertz rather than the default semitone units.

–r use least squares linear regression rather than the robust least of median squared residuals linear regression in determining the location of turning–points.

SEE ALSO

auto_prosody (PCB) – automatic prosodic transcription.
smoother (PCB) - a non–linear smoothing algorithm for Audlab parameter tracks and sample data.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

P.C. Bagshaw, "An investigation of acoustic events related to sentential stress and pitch movements, in English," Proceedings of the 4th Australian International Conference on Speech Science and Technology, Brisbane (1992).

BUGS

None known.

AUTHOR

Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai
Seika-cho
Kyoto 619-02
Japan.

NAME

pm_freq – pitch mark data to Xmg–format F0 contour

SYNOPSIS

pm_freq *pitchmark_file xmg_file*

[**–f** *sampling frequency*]

[**–l** *lower F0 value*]

[**–u** *upper F0 value*]

[**–c** *line colour*]

DESCRIPTION

A list of pitch mark times are input from *pitchmark_file*. Each time is assumed to be in milliseconds from the start of the original sample data file. Pitch mark times must be listed in chronological order.

Each time is considered in succession. The duration between consecutive pitch mark times is calculated. This duration is converted to Hertz. If the value is greater than or equal to [–l] *lower F0 limit*, and less than or equal to [–u] *upper F0 limit*, it is taken to represent the fundamental frequency at the time in the centre of the two pitch mark times. Otherwise, the duration between the marks in considered to correspond to an unvoiced region of speech, and an equals character (=) is inserted to form a break in the resultant F0 contour.

The F0 values at each each time calculated (and necessary break characters) are written to *xmg_file*. The *xmg_file* is given a header describing its contents as a list of segments connected by a solid line coloured [–c] *line colour*.

OPTIONS

pitchmark_file specifies *pitchmark_file* as the input. The file must contain a list of pitch mark times in ascii format with one value per line, and no pre–data file header. Times are assumed to be in milliseconds.

xmg_file specifies *xmg_file* as the output. This will be an Xmg headed file.

–f *sampling frequency* set the sampling rate of the original data samples from which pitch marks were made, to *sampling frequency* kHz. The default value is 20 kHz.

–l *lower F0 value* set lower limit of fundamental frequency to *lower F0 value*. Default value is 40 Hz.

–u *upper F0 value* set upper limit of fundamental frequency to *upper F0 value*. Default value is 400 Hz.

–c *line colour* set the colour of the contour when drawn using Xmg to *line colour*. The default colour is black.

SEE ALSO

pmlar (PCB) – a pitch marker for laryngograph data

DIAGNOSTICS

The diagnostics produced by PM_FREQ are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

pmlar – a pitch marker for laryngograph data

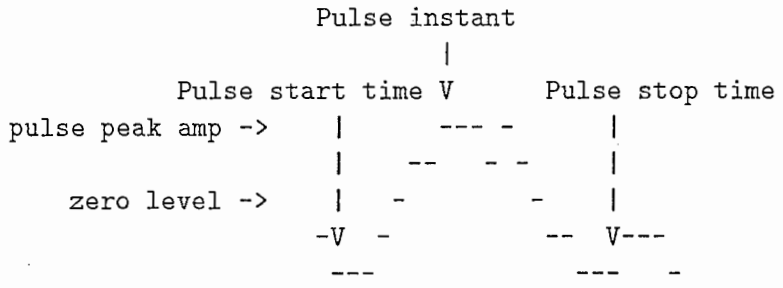
SYNOPSIS

pmlar –**i** *laryngograph data file* –**o** *output file*
[–**f** *sampling frequency*]
[–**w** *glottal pulse width*]
[–**a** *glottal pulse peak amplitude*]
[–**r**]

DESCRIPTION

The pitch marker determines the location times of glottal closure from a file of laryngograph data whose name is passed via the [–i] flag. The data must contain a pulse at each closure. The laryngograph data is assumed to contain a positive–going pulse at the instant of each glottal closure. If the input file contains pulses with negative–going pulses as markers of glottal closure instant, it may be inverted by passing the [–r] flag.

A pulse is identified for data satisfying the following description:



The 'pulse start time' is the first sample for which the amplitude is less than zero and less than or equal to the amplitude of following samples. The 'pulse stop time' is the last sample for which the amplitude is less than zero and less than or equal to the amplitude of preceding samples. The 'pulse width' is defined as the difference between the 'pulse stop time' and the 'pulse start time'. The 'pulse peak amp' is the maximum amplitude of samples between 'pulse start time' and 'pulse stop time' (always greater than zero). The 'pulse instant' is defined as the time of the first of these samples with an amplitude of 'pulse peak amp'. The 'pulse width' must be greater than [–w] *glottal pulse width*, and the 'pulse peak amp' must be greater than [–a] *glottal pulse peak amplitude.*

The input file, [–i] *laryngograph data file* must be a headerless file containing sample data of type 'short'. The rate a which samples where taken is given as [–f] *sampling frequency.* The output file, [–o] *output file* contains the list of 'pulse instant' times in terms of milliseconds from the start of the data file. Times are written in ascii format with one value per line. Each line is terminated by a '\n'.

OPTIONS

–i *laryngograph data file* specifies *laryngograph data file* as input to the pitch marker. Must be a headerless file containing sample data of type 'short'.

–o *output file* specifies *output file* as output from the pitch marker. Will be a file containing pitch mark times in milliseconds from the start of the input file, written in ascii format, and each separated by '\n'.

–f *sampling frequency* set the rate at which laryngograph data samples where taken to *sampling frequency* kHz. The default value is 20 kHz.

–w *glottal pulse width* only accept pulses with a width greater than *glottal pulse width*. This must be greater than one. The default is 4 samples.

–a *glottal pulse peak amplitude* only accept pulses with a peak amplitude greater than *glottal pulse peak amplitude*. This must be greater than zero. The default is 100.

–r invert the input data such that all positive values become negative and all negative values become positive (magnitude remaining constant).

SEE ALSO

pm_freq (PCB) – pitch mark data to Xmg–format F0 contour.

DIAGNOSTICS

The diagnostics produced by PMLAR are intended to be self–explanatory.

BUGS

The entire input data file is read into a single buffer before pulse marking is performed. On some systems, this might cause memory allocation problems.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

prosody_stats – calculate normalisation statistics for prosodic analysis

SYNOPSIS

prosody_stats –p *parameters file* –d *root directory*
–o *statistics file*
[–l *segmentation extension*]
[–f *features extension*]

DESCRIPTION

Mean and population standard deviation (p.s.d.) fundamental frequency, and the mean and p.s.d. duration and energy for each phone type are calculated from a training database for subsequent use in automatic prosodic analysis.

The [–p] *parameters file* is required to specify (at the very minimum) the track numbers which give the F0 contour and energy contour in the feature files, a string of diacritics and an Audlab format diacritic symbols file. This should be specified in the format:

:start int_params
f0_track 1
energy_track 3
:end

:start string
fn_diacritics /NFS/atrp02/p02/users/xpaul/defaults/atr_mlp.sym
merge_diacs c!~
:end

The above track values are the default. An error will result if the parameters file is of an undesirable format. The format of this file is described in further detail in the description of the program **auto_prosody**(PCB).

The training database is required to reside under the specified [–d] *root directory* path name. This directory and all subsequent sub-directories are searched in a recursive manner for pairs of similarly named files differing only by the extensions [–l] *segmentation extension* and [–f] *features extension.* The former is required to contain phonemic segmentation labels in an Audlab format (see **labelformat**(PCB)). The latter must contain the corresponding acoustic features in the form of an Audlab track file. There may be any number of such pairs in each directory. Adjacted segments in the phonemic transcription that contain the same label field are merged together if they only differ by one or more of the diacritics specified in the parameters file.

Statistics gathered from the database are written to the [–o] *statistics file.* The format of this file is compatible with that of the parameters file. The mean and p.s.d. fundamental frequency of all voiced frames given in the F0 track of all the feature files are stated as, for example:

:start float_params
mean_f0 211.692200
sd_f0 17.136303
:end

The mean and p.s.d. duration and maximum energy for each phoneme type encounted in the database are given as, for example:

:start phoneme_stats
jh 49.660805 16.644295 -14.926167 7.886746
n 60.680489 21.383167 -7.684282 2.939368
ei 100.857262 32.734207 -5.784100 2.403849
s 98.777489 31.891609 -17.578873 6.359649
ai 103.550758 42.469086 -4.412993 2.753065
r 53.130894 22.858807 -4.869702 2.665264
p 50.521591 28.466188 -16.663340 7.152871
@ 41.196014 14.940394 -7.013972 3.245027
dh 35.417137 18.905457 -10.269009 4.146455
170.959351 41.992962 -27.800053 8.648253
:end

The fields of each line give the phoneme label, mean duration, p.s.d. duration, mean maximum energy during the tenure of the phoneme, and the p.s.d. maximum energy respectively. The file is header by a series of comment lines giving the command that was used to generate the file.

OPTIONS

–p *parameters file* specifies the name of the ASCII format parameters file stipulating the track numbers of the feature files containing fundamental frequency and energy information for each utterance in the training database. The format of this file is specific to the **prosody**(PCB) software.

–d *root directory* gives the root directory path name of the training database. This directory and all subsequent sub-directories are searched for pairs of similarly named files containing phonemic segmentation files and features files in Audlab format.

–o *statistics file* the file name given specifies the destination of the statistics to be written. The output file is in ASCII and may be edited, if required. The file extension ".stats" is recommended.

–l *segmentation extension* the files in the training database with the file extension specified by this option are expected to contain a phonemic transcription in an Audlab label file format. The default extension is ".syl". The initial full stop must be included when giving the extension.

–f *features extension* the files in the training database with the file extension specified by this option are expected to contain acoustic features in an Audlab track file format. The default extension is ".trk". The initial full stop must be included when giving the extension.

SEE ALSO

auto_prosody (PCB) – automatic prosodic transcription.
labelformat (PCB) – convert between Segstat, HMM, Audlab, and Waves label file formats.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

P.C. Bagshaw, "An investigation of acoustic events related to sentential stress and pitch movements, in English," Proceedings of the 4th Australian International Conference on Speech Science and Technology, Brisbane (1992).

BUGS

The behaviour of the recursive directory search with respect to symbolic links is not known. There might be a problem if cyclic symbolic directory links have been made under the database root directory path name. So take care with your use of symbolic links.

AUTHOR

Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai
Seika-cho
Kyoto 619-02
Japan.

NAME

read_adlb_hd – read header of AUDLAB file

SYNOPSIS

read_adlb_hd *audlab_filename*

DESCRIPTION

The file *audlab_filename* is a file containing an AUDLAB header. This header is read from the file and presented to the standard output in a pleasant, readable format. The program checks that the file *audlab_filename* starts with an AUDLAB header of the correct format. If it does not, an error will occur.

SEE ALSO

alist(AUDLAB) – list contents of Audlab format file.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

rm_adlb_hd – remove header from Audlab file

SYNOPSIS

rm_adlb_hd -i *headed_file*-**o** *non-headed_file*
[–s] [–a] [–t *channel/track to select*]

DESCRIPTION

The file [–i] *headed_file* is a file containing an Audlab header. This header is removed, and the remaining contents of the file are placed into the file [–o] *non-headed_file*. The two file names passed as parameters to the program must be different. The program checks that the file [–i] *headed_file* starts with an Audlab header. If it does not, an error will occur. If the header describes the input file as containing either sample or track data, a specific channel or track may be selected using the [-t] flag. The default is to transfer the data from all the channels/tracks. Each data entry may be byte swapped by passing the [–s] flag. Entries are assumed to occupy two bytes for the data format SHORT, and four bytes for the data formats INT and FLOAT. The output data may be written in Ascii format with one value per line, by passing the [–a] flag; otherwise output is in a binary format. On normal exit, the program returns the sampling frequency (kHz) read from the header of a file of sample data, before being removed; otherwise zero.

If the input file contains label data, the program terminates without removing the Audlab header. The program labelformat should be used to change label files.

OPTIONS

–s byte swap (short, int or float). Useful when converting SUN data to PC DOS data.

–a write to the output file in an Ascii format rather than binary.

–t *channel/track to select* select one specific channel or track when transfering the data. This option may only be used with files containing sample or track data. The default is to transfer all the data given after the header.

SEE ALSO

labelformat (PCB) – convert between Segstat, HMM, Audlab, and Waves label file

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

sdt2trk – convert Audlab spatial domain track file to a simple track file.

SYNOPSIS

std2trk –**i** *sdt_file* –**o** *trk_file*

DESCRIPTION

The Audlab SDT file containing float data given by [–i] *std_file* is converted to the parallel track file named by [–o] *track file*. The input and output file names must differ. Each channel of the SDT file is formed into one of a number of parallel tracks. The track descriptor is set to "sdtn" where **n** is the SDT channel number (starting with 1). All the necessary track file header fields are inherited from the SDT file headers.

OPTIONS

None.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

DIAGNOSTICS

The diagnostics produced by SDT2TRK are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

smoother – a non–linear smoothing algorithm for AUDLAB parameter tracks and sample data.

SYNOPSIS

smoother –i *audlab_data_file* –o *smoother_data_file*

[–e]

[–l [–w *window length*]]

[–f *length of 1st median*]

[–d [–s *length of 2nd median*]]

[–t *channel/track to select*]

DESCRIPTION

The non–linear smoother provided may be used to iron out minor erroneous components evident in parameter tracks or (if required) to apply subtle linear smoothing.

Input data is read from [–i] *audlab_data_file* which must be an AUDLAB headed file containing either track data of type 'float' or sample data of type 'short'. Tracks or channels may be either in parallel or serial format. Non–linear smoothing is applied to all input tracks/channels unless one specific track or channel is selected by [–t] *channel/track to select*, in which case just that channel/track is smoother and the output file contains only one smoothed channel/track. Output data is written to [–o] *smoother_data_file* in serial format, in an order corresponding to that of the input data. Up to one hundred and twenty eight tracks/channels may be processed simultaneously.

The smoothing operation involves the use of a non–linear median filter of length [–f] *length of 1st median.* This may be followed by a linear hanning window of length [–w] *window length* by passing the [–l] flag. Double smoothing is applied if the [–d] flag is specified. This double smoothing involves delaying the input until the first median filter (and optional window) obtain an output, calculating the difference, applying a second non–linear smoother (and optional window) to this difference, and adding the result to the output of the first smoothing stage, which must also be delayed while the second smoothing stage is performed. The length of the second non–linear median filter is given by [–s] *length of 2nd median.*

A non–linear median filter of length M takes, as its input, M consecutive values, sorts them in order of increasing value, and gives, as its output, the middle value. Note that M is always odd. If more than fifty percent of the median filter's input values are breakers, then the output value will be a breaker.

A linear hanning window is used in this application as a smoothing filter. A window of length M takes M consecutive values and weights the n'th value by a factor h(n). The filter output is the sum of the weighted values.

$$h(n) = \frac{1}{M+1}\left(\frac{1-\cos 2\pi n}{M+1}\right) \quad for \;\; 1 \leq n \leq M \tag{1}$$

If a breaker exists in the input of the hanning window, then the output is also a breaker. However, if the [–e] flag is passed and the hanning window's input consists of less than fifty percent of breakers, then the length of the window is temporally reduced to occupy only the non-breaker input values.

The inherent delay imposed by the smoother will always be even, given that the length of any optional hanning window is odd. In order to prevent small time shifts as a result of this delay, dummy breakers are used as additional input or output values. If the [–e] flag is passed, the smoothing algorithm is initialised by inputting a number of breakers corresponding to half of the delay before reading data from the input file. Once the data of a track/channel has been read, the filter is then flushed by a number of further dummy breakers corresponding to the other half of the delay. If, however, the [–e] flag is not passed, an equal number of breakers are inserted at the start of the output file, and appended to the end of the output file.

If *audlab_data_file* contains AUDLAB tracks, then each track descriptor is appended with "_smth" in the headers of *smoother_data_file*. In all cases, the maximum and minimum values of each smoothed track/channel are recorded in the corresponding header descriptor.

OPTIONS

–i *audlab_data_file* specifies *audlab_data_file* as input to the algorithm. Must be an AUDLAB file containing track data of type 'float' or sample data of type 'short'.

–o *smoother_data_file* specifies *smoother_data_file* as output from the algorithm. Will be an AUDLAB file containing smoothed tracks/channels following one after another.

–e use dummy breakers at the beginning and end of each track/channel to compensate for filter delay, and ignore breakers in the hanning window when they are the minority of values over the window length.

–l apply linear hanning window to output of median filter(s).

–w *window length* set the length of the hanning window to be used for linear smoothing to *window length*. Must be odd, greater then two and less than one hundred and twenty eight. Default is three.

–f *length of 1st median* set the number of points to be used in the first non–linear median filter to *length of 1st median*. Must be odd, greater than two and less than one hundred and twenty eight. Default is five.

–d apply double smoothing process.

–s *length of 2nd median* set the number of points to be used in the second non–linear median filter to *length of 2nd median*. Must be odd, greater than two and less than one hundred and twenty eight. Default is five.

–t *channel/track to select* select one specific channel or track from the input file when smoothing the data. The default is to smooth all the data (every channel/track) given in the file.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

L.R. Rabiner, M.R. Sambur, and C.E. Schmidt, "Applications of non–linear smoothing algorithms to speech processing," IEEE Trans. Acoustics, Speech, and Signal Processing ASSP–23 No.6 pp.552–557 (1975).

DIAGNOSTICS

The diagnostics produced by the SMOOTHER algorithm are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

spectral_inversion – invert the spectral chacteristics of a speech waveform.

SYNOPSIS

spectral_inversion **–i** *speech file* **–o** *new speech file*
[**–r** *lower_frequency upper_frequency*]
[**–f** *sampling frequency*]
[**–p** *power of 2 transform size*]
[**–w** *frame length*]
[**–s** *frame shift*]

DESCRIPTION

The algorithm implemented performs spectral inverted speech resynthesis using fast Fourier transforms.

Segments of data of length [–w] *frame length* (ms) are taken from the speech data in [–i] *speech file*. Successive frames are shifted by [–s] *frame shift* (ms). Each frame is passed through a Blackman–Harris window and its spectrum is calculated by a Fast Fourier Transform using (2 to the power of [–p] *power of 2 transform size*)-points. The FFT size must be big enough to contain all the samples in a frame. Zero padding is used to fill excess points. The frequency bins of the FFT are inverted such that, for an N-bin spectrum, bin[i] <–> bin[i + N/2] for i=L,L+1,...,U where L and U correspond to the bins for *lower_frequency* (Hz) and *upper_frequency* (Hz) respectively, as specified by the [–r] option. L will be greater than or equal to zero and U will be less than N/2. The inverse FFT of the inverted spectrum is evaluated and the resultant waveform is passed through a Blackman–Harris window. The data from successive overlapping frames is accumulated to form a new speech waveform which is written to [–o] *new speech file*.

An integer number of complete frames are processed. This may result in the output speech data being slightly shorter then the input data. If the *lower_frequency* and *upper_frequency* are equal then no spectral inversion takes place but each frame is still transposed through the frequency domain. Automatic gain control is used to ensure that the variation in amplitude in the output signal is approximately the same as that for the input signal.

OPTIONS

–i *speech file* specifies *speech file* as input to the algorithm. Must be a headerless file containing sample data of type 'short'.

–o *new speech file* specifies *new speech file* as output from the algorithm. This will be a headerless file containing the spectal inverted speech waveform.

–r *lower_frequency upper_frequency* defines the range of FFT frequency bins (Hz) for which the spectrum is to be inverted. The *lower_frequency* must be a positive

value. The *upper_frequency* must be greater than *lower_frequency* and less than half the data sampling frequency. The default range is from 100.0Hz to 4500.0Hz.

–f *sampling frequency* specifies the rate at which the original speech was sampled at as *sampling frequency*, in Hertz. Default is 20000Hz.

–p *power of 2 transform size* set the number of points to be used in the FFT to two to the power of *power of 2 transform size*. Must be greater than five. Default setting is 9 (a 512–point FFT).

–w *frame length* set the length of the analysis frame to *frame length* in milliseconds. Must be greater than zero. Default frame length is 20.0ms.

–s *frame shift* set the shift between consecutive analysis frames to *frame shift* in milliseconds. Must be greater than zero. Default frame shift is 5.0ms.

DIAGNOSTICS

The diagnostics produced by SPECTRAL_INVERSION are intended to be self–explanatory.

BUGS

None known.

AUTHOR

Paul Bagshaw
ATR Interpreting Telephony Research Laboratories
Hikaridai
Seika-cho
Kyoto 619-02
Japan.

NAME

srpd – a super resolution pitch determination algorithm (Sun Version)

SYNOPSIS

srpd –i *audlab sample file*
[–o *pitch file* [–F] [–P] [–W –T | –a]]
[–c *cross correlation coeff file*]
[–l *lower pitch frequency limit*]
[–u *upper pitch frequency limit*]
[–d *decimation factor*]
[–n *noise floor*]
[–h *unvoiced to voiced coeff threshold*]
[–m *min voiced to unvoiced coeff threshold*]
[–r *voiced to unvoiced coeff threshold-ratio*]
[–t *anti pitch doubling/halving threshold*]
[–p]
[–s *frame shift*]
[–w *artificial frame length*]

DESCRIPTION

See (Medan, Yair, and Chazan, 1991) and (sections 2.6 & 4, Bagshaw, 1991) for a detailed description of the algorithm.

Frames of data are read in from [–i] *audlab sample file* in chronological order such that each frame is separated from its predecessor by [–s] *frame shift* (ms). To allowing the output data to be synchronised with other signal processing algorithms such as cepstral analysis and formant tracking, the [–w] *artifical frame length* may be specified. This ensures that an equal number of frames are analysed by difference signal processing algorithms. Each frame is analysed in turn.

The maximum absolute signal amplitudes are initially found over the duration of two segments, each of length N_min samples. If the sum of their absolute values is below two times [–n] *noise floor*, the frame is classified as representing silence and no coefficients are calculated (all set to zero). Otherwise, a cross correlation coefficient p(n) is calculated for all n from a period in samples corresponding to [–u] *upper pitch frequency limit* (N_min) to a period in samples corresponding to [–l] *lower pitch frequency limit* (N_max), in steps of [–d] *decimation factor*. For values of n for which p(n) is not calculated in this range, p(n) is set to zero. p(n) is considered to be invalid if the number of zero crossings over the first two segments (2n samples) is less than four, ie. if two pitch periods do not reside within the segments. In calculating the coefficient p(n) only one in [–d] *decimation factor* samples of the two segments are used. Such down-sampling permits rapid estimates of the cofficient

p(n) to be calculated over the range N_min n N_max. This results in a cross-correlation track for the frame being analysed.

Local maxima of the track with a coefficient value above a specified threshold form candidates for the fundamental period. The threshold is adaptive and dependent upon the values [–h] *unvoiced to voiced coeff threshold*, [–m] *min voiced to unvoiced threshold*, and [–r] *voiced to unvoiced coeff threshold–ratio*. If the previously analysed frame was classified as unvoiced or silent (which is the initial state) then the threshold is set to [–h] *unvoiced to voiced coeff threshold*. Otherwise, the previous frame was classified as being voiced, and the threshold is set equal to [–r] *voiced to unvoiced coeff threshold–ratio* times the cross-correlation coefficient value at the point of the previous fundamental period in the former coefficients track. This product is not permitted to drop below [–m] *min voiced to unvoiced coeff threshold*.

If no candidates for the fundamental period are found, the frame is classified as being unvoiced. Otherwise, the candidates are further processed to identify the most likely true pitch period. During this additional processing, a threshold given by [–t] *anti pitch doubling/halving threshold* is used.

With the optional flag [–p] passed, biasing is applied to the cross-correlation track as described in (Section 4.2, Bagshaw, 1991).

If a frame is classified as being silent or unvoiced, a pitch value of zero is placed in the tracks "srpdf0" and "srpdper"; otherwise, the determined pitch value is given.

OPTIONS

The default values given below were found to optimise the performance of the pitch determination algorithm for speech data sampled at 20kHz using a 16–bit ADC and low pass filter with a 600Hz cut–off frequency and more than –85dB rejection above 700Hz. The best performances occur if the [–p] flag is passed.

–i *audlab sample file* specifies *audlab sample file* as input to the algorithm. The file must be an AUDLAB headed file containing sample data of type 'short'.

–o *pitch file* specifies *pitch file* as output from the algorithm. This is an AUDLAB file containing one or more (up to four) interleaved tracks (ie. in parallel format). The possible tracks (in order) describe the fundamental frequency (Hz), the pitch period (ms), the maximum cross–correlation value, and the adaptive threshold value for each frame analysed. The [–F] flag and/or [–P] flag and/or [–W] and/or [–T] flag must also be passed.

–F include a track of frame pitch frequencies (Hz) in *pitch file*. This track is labelled as "srpdf0" in the AUDLAB file headers.

–P include a track of frame pitch periods (ms) in *pitch file*. This track is labelled as "srpdper" in the AUDLAB file headers.

–W include an F0 reliablity track in *pitch file* containing a weight ranging from minus one to plus one for each frame analysed. The weight for each frame is given by the adaptive threshold value subtracted from the maximum cross-correlation value for that frame, scaled by a factor equal to one minus the threshold if the difference is positive (voiced), and equal to one plus the threshold if it is negative (unvoiced). This track is labelled "srpd_weight" in the AUDLAB file headers. This flag cannot be passed in conjunction with the [–a] flag.

–T include two tracks in *pitch file* containing the maximum cross-correlation value and the adaptive threshold value for each frame analysed. These tracks are labelled "srpd_maxcoeff" and "srpd_threshold" respectively in the AUDLAB file headers. This flag cannot be passed in conjunction with the [–a] flag.

–a force *pitch file* to be absent of an AUDLAB header and each data entry to be represented in ascii format. Only applicable when the [–o] flag is passed. When this flag is passed, only one of the tracks specified by the [–F] or [–P] flag can be made, not both. This flag cannot be passed in conjunction with the [–W] flag or [–T] flag.

–c *cross correlation coeff file* *cross correlation coeff file* will contain the cross correlation coefficients obtained for each segment of each frame analysed, in the form of an AUDLAB spatial domain track.

–l *lower pitch frequency limit* set the lowest pitch frequency allowed to *lower pitch frequency limit*. Must be greater than zero. Default is 40.0Hz.

–u *upper pitch frequency limit* set the highest pitch frequency allowed to *upper pitch frequency limit*. Must be greater than current *lower pitch frequency limit*. Default is 400.0Hz.

–d *decimation factor* set down-sampling for quicker computation so that only one in *decimation factor* samples are used in the first instance. Must be in the range of one to ten inclusive. Default is four.

–n *noise floor* set the maximum absolute signal amplitude that represents silence to *noise floor*. If the absolute amplitude of the first segment in a given frame is below this level at all times, then the frame is classified as representing silence. Must be a positive number. Default is 120.

–h *unvoiced to voiced coeff threshold* set the correlation coefficient threshold which must be exceeded in a transition from an unvoiced classi-

fied frame of speech to a voiced frame as *unvoiced to voiced coeff threshold*. Must be in the range zero to one inclusive. Default is 0.88.

–m *min voiced to unvoiced coeff threshold* set the minimum allowed correlation coefficient threshold which must not be exceeded in a transition from a voiced classified frame of speech to an unvoiced frame, as *min voiced to unvoiced coeff threshold*. Must be in the range zero to *unvoiced to voiced coeff threshold* inclusive. Default is 0.75.

–r *voiced to unvoiced coeff threshold–ratio* set the scaling factor used in determining the correlation coefficient threshold which must not be exceeded in a voiced frame to unvoiced frame transition, as *voiced to unvoiced coeff threshold–ratio*. The voiced to unvoiced coefficient threshold is determined by multiplying this scaling factor with the maximum cross-correlation coefficient of the previously voiced frame. If this product is less than *min voiced to unvoiced coeff threshold* then this is used instead. Must be in the range zero to one inclusive. Default is 0.85.

–t *anti pitch doubling/halving threshold* set the threshold used in eliminating (as far as possible) pitch doubling and pitch halving errors as *anti pitch double/halving threshold*. Must be in the range zero to one inclusive. Default is 0.77.

–p perform post–processing and apply coefficient bias on most likely pitch region.

–s *frame shift* sets the time difference between each analysed frame to *frame shift* in milliseconds. Default is 6.4ms.

–w *artifical frame length* sets the frame length required to synchronise the output tracks with those generated using other signal processing algorithms to *artifical frame length*. Default is 0.0ms.

SEE ALSO

srpd (S) – a super resolution pitch determination algorithm (PC Version)
srpd_hd – a super resolution pitch determination algorithm for headerless data (Sun Version)
srpd_old (PCB) – an unmodified super resolution pitch determination algorithm

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.

Y. Medan, E. Yair, and D. Chazan, "Super resolution pitch determination of speech signals," IEEE Trans. Signal Processing Vol.39 No.1 pp.40–48 (1991).

P.C. Bagshaw, "On the determination of fundamental frequency in speech signals for the analysis of prosody," Centre for Speech Technology Research, Edinburgh, 1991.

DIAGNOSTICS

The diagnostics produced by the SRPD algorithm are intended to be self-explanatory.

BUGS

None Known.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN

NAME

xmg_adlb – convert a Xmg format segment file to an Audlab headed track file.

SYNOPSIS

xmg_adlb *xmg_file audlab_file*

[–y *y scale multiplier*]

[–b *break number*]

[–s *shift*]

[–l *audlab file lower limit*]

[–u *audlab file upper limit*]

DESCRIPTION

The line segment data in the Xmg format file called *xmg_file* is transferred to Audlab format in the file named *audlab_file* with an appropriate header. Values in the Audlab file are interpolated from the times and values given in the Xmg file at time shifts given by [–s] *shift* (ms). The breaks between line segments are represented in the output file by [–b] *break number*. The interpolated input data values are multiplied by [–y] *y scale multiplier* before being entered into the output file.

OPTIONS

–y *y scale multiplier* set the multiplication factor which is applied to the input data to *y scale multiplier*. This must be an integer value and has the default of one.

–b *break number* set the breaker value to *break number*. Default is 0.0.

–s *shift* set the duration between each interpolated value to *shift* (milliseconds). The default value is 5.0 ms.

–l *audlab file lower limit* set the default display lower data limit to *audlab file lower limit*. The default is to inherit the minimum y–axis level set in the Xmg file. If, however, the [–u] flag is used, the default value becomes 0.0.

–u *audlab file upper limit* set the default display upper data limit to *audlab file upper limit*. The default is to inherit the maximum y–axis level set in the Xmg file. If, however, the [–l] flag is used, the default value becomes 0.0.

DOCUMENTATION

Audlab User Manual Ver3.1. CSTR., University of Edinburgh.
XMG, a multiple graph drawer for X11.

DIAGNOSTICS

The diagnostics produced by XMG_ADLB are intended to be self–explanatory.

BUGS

The program only converts Xmg files containing line segments.

AUTHOR

Paul Bagshaw
Centre for Speech Technology Research
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN