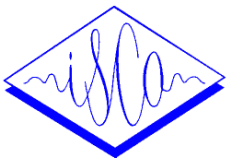

3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

A PITCH CONTOUR ANALYSIS GUIDED BY PROSODIC EVENT DETECTION

Edouard Geoffrois*

NTT Human Interface Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo 180, Japan

ABSTRACT

A left-to-right algorithm for analyzing pitch contours and estimating their underlying representation as a sequence of parametered commands is proposed. It is based on a generalization of recursive least squares optimization to a nonlinear model. This parameter fitting is embedded in a search guided by prosodic event detection, which determines when a candidate structure has to be augmented to keep fitting incoming data. The algorithm applied to Japanese read sentences successfully estimated 91% of lexical pitch accent positions.

Keywords: prosody, intonation, speech recognition.

1 INTRODUCTION

Though using prosodic information in speech recognition systems is a long-standing goal, the powerful methods developed for speech recognition do not apply straightforwardly to this kind of information. Among the strongest impediments are its suprasegmental nature and its high variability. Though prosody has received much attention from the linguistic point of view and for speech synthesis, little literature is available for speech recognition.

A recently proposed approach facilitates the integration of prosodic analysis into a recognition system [5]. Using an analysis-by-synthesis method, a sequence of prosodic features is first synthesized for each candidate yielded by the recognition, these predicted features are then compared with observed features, and the resulting score is combined with sentence likelihood to reorder the candidates. One advantage of performing prosodic analysis after phonemic recognition completion is that phoneme boundaries are available for computing durational features. Pitch information, however, still requires complex preprocessing to be incorporated in such a scheme, and therefore deserves specific investigation.

After explaining how the prosodic nature of pitch contours can be dealt with, an algorithm implementing the proposed approach and its application to Japanese utterances will be presented in sections 3 and 4.

*On leave from LIMSI, BP 133, F-91403 Orsay cedex, France.
E-mail: geoffroi@limsi.fr

2 ANALYZING PITCH CONTOURS

Pitch results from the interaction of overlapping phenomena with various time spans, ranging from local microprosodic effects to long-term trends such as declination. Such long-range interactions prevent local interpretation of pitch values (which is the idea expressed by the term suprasegmental). Therefore, a mere concatenation of context-independent patterns does not provide an adequate representation. Moreover, a formal grammar cannot fully model these phenomena without additional quantitative knowledge, especially about time course. An interface between symbolic (phonemic) and surface (phonetic) structures is therefore needed. This intermediate layer should be a quantitative model in which pitch values are expressed as a function of the whole underlying structure rather than of a single symbol.

The Fujisaki model provides such an intermediate layer for Japanese and other languages [1]. It represents a pitch contour by an underlying sequence of two types of commands, namely the phrase commands representing utterance components and the accent commands representing lexical accent components. Commands drive second-order linear filters whose outputs are added to yield F_0 values. The effect of different commands can thus widely overlap. This allows the model to accurately fit Japanese read speech contours with only a few linguistically motivated commands. Overlapping, however, makes estimation of commands from pitch contours more difficult. Specifically, a new command must be estimated conditionally to the commands already found. This prevents from using efficient search algorithms such as dynamic programming or Viterbi search. Furthermore, whereas a phoneme is classically constrained to begin right after the end of the previous phoneme, temporal constraints on commands are less clear and the search space is not well defined. To deal with those problems, an event detection scheme has been introduced [3], which is the subject of section 3.2.

A second characteristic of pitch movements is their variability. Independently of contextual effects, the same linguistic phenomenon can be expressed by various movements with a fairly wide range of amplitudes and dura-

tions. These form a continuum rather than different categories, or at least categories with very fuzzy boundaries. For example, pitch deviations on a stressed syllable in English can have various amplitudes, and deciding whether a given deviation is large or small is mainly a matter of subjective evaluation. A parametered model like the Fujisaki model deals with this variability by representing different but equivalent pitch contours by the same underlying structure with different parameter values.

During the search, parameters adjustment and structure estimation have to be tightly coupled. Indeed, as mentioned above, the introduction of a new command is highly dependent upon previously estimated commands. The stability of the search cannot be ensured if estimation errors are allowed to propagate. Parameters must therefore be constantly reoptimized. An efficient parameter optimization algorithm (section 3.1) has been designed to perform the whole analysis in reasonable time.

A third impediment for analyzing pitch contours is that no data is available during unvoiced parts. The modeled contour therefore cannot be constrained in those parts. In fact, the discrimination between voiced and unvoiced segments is not all-or-none. A measure of voicing computed during pitch extraction can be used to enforce a tighter fit on vocalic nuclei, while a looser fit is granted on less reliable parts. This is done by weighting the criterion to minimize for parameter fitting. Setting weights to zero in unvoiced parts is a simple and suitable way to deal with the absence of data.

Finally, the algorithm presented in the next section deals with the prosodic nature of pitch by (1) allowing a single structure to represent various pitch contours and using a least squares optimization to fit modeled contours to the voiced parts of the observed contours and (2) performing a search guided by event detections.

3 ALGORITHM

3.1 Recursive Least Squares Optimization

Using the functional formulation given by Fujisaki, F_0 is expressed as a function of time which depends on several parameters, such as command timings, amplitudes, and time constants. Between two event detections, the number of commands stays the same, and so does the number of parameters. Let n be this number, $\Lambda = (\lambda_1, \ldots, \lambda_n)$ the parameter set, and $f(\Lambda, t)$ the function expressing the modeled contour. The problem is to keep the parameter values such that the distance between the modeled contour and the observed contour remains minimal when further data is taken into account. This problem can be defined as the minimization of the following energy function:

$$E(\Lambda, t_0) = \frac{1}{2} \sum_{t=0}^{t_0} w(t)(f(\Lambda, t) - p(t))^2 \qquad (1)$$

where $p(t)$ is the observed pitch. Pitch was extracted with the lag-window method [4]. A measure a voicing was simultaneously computed and used as the weight $w(t)$ to get a closer fit on reliable segments. This measure was given by the height of the peak used to compute F_0 value by peak picking. When pitch contours are preprocessed to select reliable movements (defined as a coherent sequence of F_0 values for which the cumulated voicing measure overcomes a threshold), weights are set to zero outside the selected parts. Minimizing the energy function is equivalent to solve the following n equations expressing that derivatives respective to all parameters are zero:

$$\sum_{t=0}^{t_0} w(t) \frac{\partial f}{\partial \lambda_i}(\Lambda, t)(f(\Lambda, t) - p(t)) = 0 \qquad (2)$$
$$\forall i = 1, \ldots, n$$

The sum extends up to time t_0. When t_0 increases, the position of the minimum changes and new parameter values has to be computed. Assuming that the minimum is already found for time $t_0 - 1$, the new minimum can be found as a perturbation of the previous one using a first-order development. Let Λ^* be the optimal parameter set for time $t_0 - 1$, and let the new optimal parameter values be written as $\lambda_i = \lambda_i^* + \Delta\lambda_i$. First-order development around Λ^* of equations 2 and subtraction of the recursion hypothesis (same equations at time $t_0 - 1$ for Λ^*) yield the following recursion formula:

$$\sum_{j=1}^{n} a_{i,j}(t_0)\Delta\lambda_j \approx b_i(t_0) \qquad (3)$$
$$\forall i = 1, \ldots, n$$

where

$$a_{i,j}(t_0) = a_{i,j}(t_0 - 1) + w(t_0)\frac{\partial f}{\partial \lambda_i}(\Lambda^*, t_0)\frac{\partial f}{\partial \lambda_j}(\Lambda^*, t_0) \qquad (4)$$

$$b_i(t_0) = w(t_0)(p(t_0) - f(\Lambda^*, t_0))\frac{\partial f}{\partial \lambda_i}(\Lambda^*, t_0) \qquad (5)$$

These equations allow to compute the $\Delta\lambda_j$ by inverting a matrix and thus get the new minimum recursively. The matrix itself is built recursively too.

The full derivation of equations 3, comments on the approximations involved and on stability, introduction of analog constraints, and implementation details can be found in [2].

3.2 Prosodic Event Detection

The principle of event detection is to detect new information in the incoming data, i.e. data unpredictable from the structure estimated up to that point. This is why optimization must occur before detection: no new information should be found if data can be fitted with the existing structure.

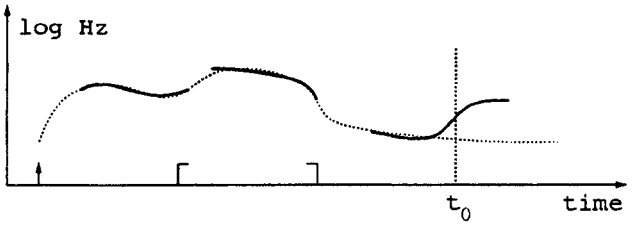

Figure 1: Principle of event detection. At time t_0, the predicted contour (dotted line) is the optimized contour corresponding to the underlying command sequence estimated so far (on the horizontal axis: one upward phrase command, an accent command onset and end). When incoming data (bold line) deviates from the predicted contour, that means it bears new information. An event is detected when the discrepancy becomes significant, and new commands corresponding to the deviation are introduced at this point in time.

The structure representing the observed contour is built gradually, the structure being augmented with new commands when an event is detected. Using the optimization process explained above, a given structure is constantly adapted to fit the observed contour. When a discrepancy between the fitted and the observed contours appears in the last few frames, that indicates that the structure cannot represent the observations successfully, and that the structure must therefore be augmented with one (or possibly more) new command. It is the apparition of such a discrepancy showing that incoming data bears new information that we call an event. In the present case, an event is either an upward or a downward pitch trend. Introducing new commands only when necessary makes the search efficient even though the search space is huge.

Event detection is performed by simply thresholding a weighted average of the difference between the observed and the modeled pitch over the last few frames. This averaging prevents an event from being triggered before a clear trend is found. The average is implemented as the output of a second-order filter applied to the framewise difference weighted by the pitch extraction reliability measure. To prevent multiple triggering of the same event if the average fluctuates around the threshold, detection occurs only if the threshold has not already been overcome in the last few milliseconds. This refractory period was set to 60 ms.

Another interpretation of event detection can be given by considering energy landscapes. When more data is taken into account ($t_0 \rightarrow t_0 + 1$), the landscape is slightly modified, and the recursive optimization keeps track of the minimum. Insofar as the candidate structure fits the observations, augmenting that structure cannot lower the minimum. But if a modified structure can provide a better fit, another valley appears in the energy landscape, and search has to proceed along that valley. The purpose of event detection is thus to determine when an alternative structure can give a better fit, i.e. when a local minimum splits, so that the search keeps track of all local minima.

3.3 Search

The aim is to find the structures for which the optimized criterion is smallest. For that purpose, a beamsearch is performed and the best hypotheses are kept at each frame.

When an event is detected, the structure is augmented according to the type of event and to the preexisting structure, within grammatical constraints [3, Table 1]. For each possible command or set of commands, a new hypothesis is created for the corresponding augmented structure. The original hypothesis is not systematically discarded because of possible false alarms in event detection. Therefore, the number of hypotheses increases when an event is detected. This number is limited by the beamwidth before proceeding to the next frame.

The resulting algorithm is as follows. For each frame, each hypothesis is reoptimized. It is then checked for event detection and new hypotheses are created accordingly. All hypotheses are then reordered, and the top-ranked ones are kept for the next frame.

4 RESULTS

Fifty-one Japanese sentences uttered by one speaker were analyzed. The sentences were uttered in a rather spontaneous manner, but without hesitations, and with marked pauses between breath groups.

The contours were hand-labeled with lexical pitch accent information, both by looking at them and by listening. Labels were put on pitch rises and falls reflecting accent type and accent position. Syntactical information, as represented by phrase commands in the Fujisaki model, was not as easy to determine because the sentences were uttered as sequences of independent groups, and was therefore not labeled. Rather, a rule was used, according to which an upward phrase command was expected at the beginning of each group, and a downward phrase command was expected at its end except when the last word of the group was unaccented. There were 184 groups, and 492 accent labels.

Analysis was performed with a beamwidth of 10, and the 10 top candidates were memorized. Examples of candidate structures are shown in Figure 2. For each sentence, only the best two candidates were used for evaluation.

Differences between estimated and expected command sequences fell into various cases which were gathered into four broad categories: (1) insertion and deletion of accent commands, (2) insertion and deletion of phrase commands. (3) misrepresentations of unaccented words and of silent parts, and (4) timing errors. A more detailed description can be found in [2].

These differences had several causes: (a) estimation errors, i.e. failure of the algorithm to find the optimal structure; (b) lack of using complementary information such as intensity; (c) inappropriate modeling, especially of unaccented word; (d) equivalent representations. These results pinpointed the constraints further needed to improve accuracy.

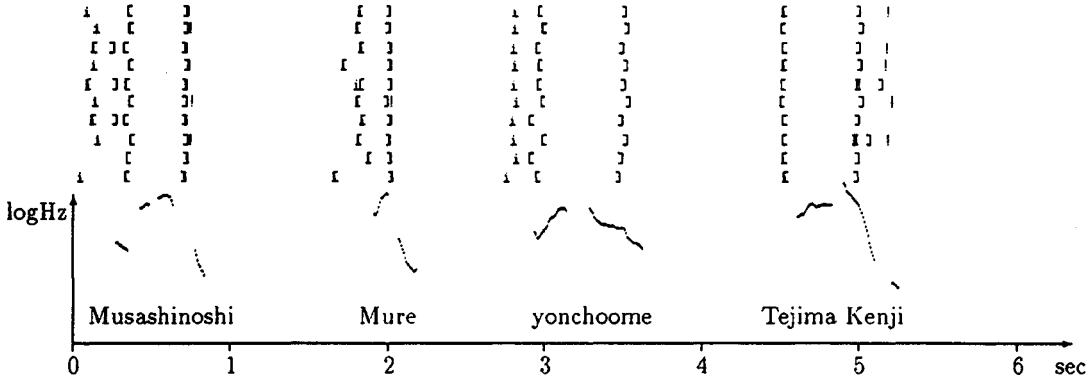

Figure 2: Example of analysis. The utterance is a request for a subscriber's phone number. It is made of four groups. The first three are the city name, district name and block number, respectively. The last group is the name of the subscriber. The block number ('yonchoome') is unaccented: the pitch rises sharply at the beginning then goes down slowly. The three other groups are accented words: there is a clear pitch fall which defines the accent position. Candidate structures ranked first (top) to 10^{th} (bottom) are shown above the input contour (accent command onset and end, and upward and downward phrase commands, are represented by the characters [,], i, and !, respectively). The first candidate has exactly the expected structure, with an upward phrase command before each group and a downward phrase command after each group except the unaccented one. Furthermore, the accent command found on the 'unaccented' group had a much smaller amplitude then the other ones (not illustrated). Other candidates are variants of the first one, and exemplify typical errors, such as inserted accent commands (at the beginning of 'Musashino' and at the end of 'Tejima Kenji'), deleted phrase commands (same locations), and early timing (before 'Mure' in the 4^{th} and 10^{th} candidates).

Only differences of types a, b and c were considered as errors. The total number of errors for accent commands was 45. Therefore, 91% of labeled accent commands were successfully estimated.

5 CONCLUSION

A model-based search using parameter optimization and event detection was proposed for analyzing pitch contours. It yields the estimated structure of a contour, along with the optimal parameter values. Parameter optimization provides an interface between underlying structures and highly variable observed contours. Event detection deals with a context-dependent model and a huge search space. The results show that prosodic information can be extracted from F_0 contours without the need of a priori information such as segmentation. This can be used to constrain a Japanese speech recognition system with lexical accent information.

Integration of other sources of information, however, should improve accuracy. This is clear for intensity. Word and phoneme boundaries should also lessen ambiguities. Conversely, integration in a recognition system with the analysis-by-synthesis approach can benefit from the proposed parameter optimization method. Rather than directly checking the feature sequence synthesized from recognition output against observed features, parameter adaptation provides an interface for handling contour variability. Moreover, the optimal parameter values are available as additional information. This research thus provides methods for using pitch in speech recognition systems.

Acknowledgements

This work was initiated during a traineeship at NTT Basic Research Laboratories, matured while in LIMSI and in Mary Ostendorf's team at Boston University, and was eventually implemented and evaluated at NTT Human Interface Laboratories. I gratefully acknowledge the valuable help and advice, the nice working environment and the encouraging atmosphere I have found in each of these places. I especially wish to thank all members of the Furui Research Laboratory for their kind support.

References

[1] H. Fujisaki. The role of quantitative modeling in the study of intonation. In *International Symposium on Japanese Prosody*, pages 163–174, 1992.

[2] E. Geoffrois. Estimation of prosodic events from Japanese F_0 contours. In *Technical Report of IEICE*, June 1993.

[3] E. Geoffrois. Prosodic event detection from F_0 contours using the Fujisaki model. In *Spring Meeting of the Acoustical Society of Japan*, pages 187–188, Mar. 1993.

[4] S. Sagayama and S. Furui. Pitch extraction using the lagwindow method. In *IEICE Meeting*, pages 5–263, 1978 (in Japanese).

[5] C. W. Wightman, N. M. Veilleux, and M. Ostendorf. Use of prosody in syntactic disambiguation: An analysis-by-synthesis approach. In *DARPA Workshop on Speech and Natural Language*, pages 384–389, Feb. 1991.