SPEAKER-SPECIFIC PITCH CONTOUR MODELING AND MODIFICATION

David T. Chappell and *John H. L. Hansen*

Robust Speech Processing Laboratory
Duke University, Box 90291, Durham, NC 27708-0291
http://www.ee.duke.edu/Research/Speech d.chappell@ieee.org jhlh@ee.duke.edu

ABSTRACT

This paper describes new techniques for modeling and generating speaker-dependent pitch contours for sentences. Speech synthesis applications could generally benefit from such speaker-specific pitch contours. The proposed algorithms begin with an existing pitch contour for an utterance and use data from training utterances to modify the contour to be appropriate for a second speaker. One approach modifies the original pitch values to statistically match the desired speaker at each point in time. A second novel approach uses dynamic time warping (DTW) to select a new pitch contour from a pre-determined code book and time-align the chosen contour to the original sentence. Such contour mapping can transfer one speaker's natural pitch characteristics to another person's speech. Informal listener evaluations suggest that while shifting the frequency range of the original pitch contour yields some improvement, better results are obtained by applying DTW techniques to time-warp the contour from an existing sentence produced by the desired speaker.

1. INTRODUCTION

The intonation of a sentence or phrase is represented by changes in the pitch pattern. In English, intonation varies as a function of stress and can convey emotions such as anger and surprise. Pitch can control whether an utterance is interpreted as a statement or a question. Furthermore, pitch variations impart varying degrees of lexical stress at the syllable and phrase level. Thus, the pitch contour helps convey the message of a spoken communication. Sentences in spoken English can be divided into intonational units known as tone groups or intonational phrases. Tone groups indicate afterthoughts, scope, or restrictions and are often marked by commas in written text [4, 9].

An appropriate pitch structure is important for the perceived naturalness of synthesized speech. There are several models for generating pitch contours for text-to-speech (TTS) systems. Modern TTS systems include rules for sentence-level pitch changes and also account for the articulation of speech segments. These algorithms construct an intonation contour according to the sentence type and therefore predict how pitch changes with syntactic structure, lexical stress pattern, rhythmic position, and emphasis [2, 3, 7, 8]. In addition, some TTS systems go beyond standard rule-based pitch generation and use automatic data-driven modeling [5].

We hypothesize that intonation contours carry speaker traits, and thus an accurate speaker-dependent contour is important for speaker-specific forms of speech synthesis such as waveform concatenation. Speakers of different languages and dialects apply different intonation patterns, and individual speakers may also apply rules in unique ways. Psychoacoustic experiments support the theory that F_0 contours contain speaker individualities [1].

This study proposes several algorithms for modifying the pitch contour of a sentence for the sake of imparting perceptually important characteristics of a desired speaker. We start with a pitch contour from reference data or a reference speaker for the sentence under test (S.U.T.) and use the contours of known training sentences to shape a contour for the utterance which is customized to the desired speaker. These algorithms require a set of sentences to have been collected and phonemically labeled for both the reference and desired speakers. The result of each algorithm is a new pitch contour which may be applied to the S.U.T. A synthesis system could use general rules to produce a generic pitch contour and then use one of the proposed algorithms to modify the generic contour into one which contains speaker-specific structure so that the final voice will be recognized as the desired speaker.

2. ALGORITHM DESCRIPTIONS

This section presents three algorithms for speaker-dependent pitch contour modeling and generation. Table 1 shows an overview of the techniques. For the first two algorithms, statistics are used to form a mapping function from a reference speaker's pitch frequency to the appropriate value for a second speaker. This approach works from the paradigm of a one-to-one mapping of the pitch frequencies between any two speakers. In contrast, the third algorithm employs dynamic time warping (DTW) to estimate the mapping of pitch contours between sentences. The algorithm based on DTW principles selects and time-warps a pitch contour according to known sentence-level contours for the speakers. Once the desired pitch contour has been generated, the Pitch-Synchronous Overlap and Add (PSOLA) algorithm is used to adjust the pitch [6].

For each algorithm, we present an informal subjective evaluation and discuss the strengths and weaknesses. Although the major intention of this study is to suggest prosody adjustment algorithms for speaker-dependent speech synthesis, we demonstrate the pitch generation algorithms using natural test utterances to avoid the artifacts

Algorithm	Approach	Assumptions	Main Idea
Gaussian	statistical	Gaussian distribution	Gaussian normalization
Scatterplot	statistical	independence of segments	estimate pitch-mapping function
Code book	DTW	model sentence contours exist	use existing sentence contours as models

Table 1: Overview of algorithms

inherent in existing synthesis systems. Sentences from the TIMIT database were used in this evaluation. The listener test included both a set of seven speakers who had seven sentences in common and several pairs of speakers who had two sentences in common.

2.1. Gaussian Normalization

The first algorithm uses Gaussian normalization to perform a mapping from the reference pitch values to the desired frequencies. The sample mean and sample standard deviation of the pitch are calculated for each speaker, and then pitch frequencies are translated from one speaker to another by assuming a Gaussian distribution.

Algorithm: **(A)** Perform pitch tracking of training sentences from both the reference and desired speakers. **(B)** Collect data on pitch values for each speaker. Estimate the mean and standard deviation of pitch for each speaker. **(C)** For the S.U.T., use Equation 1 to convert each original pitch value to a new frequency on a frame-by-frame basis.

The reference speaker's pitch statistics are projected to match the mean and variance of a desired speaker via the equation

$$x_2 = \frac{x_1 - \mu_1}{\sigma_1} \cdot \sigma_2 + \mu_2 \qquad (1)$$

where μ and σ represent the mean and standard deviation respectively. While one could use only the mean, including the standard deviation yields improved accuracy.

Results: Figure 1 shows the original and modified pitch contours when this algorithm is applied to the sentence, "Don't ask me to carry an oily rag like that." In this example, the reference and desired speakers have a similar high pitch range but a different low pitch range, hence the difference in the modification of the pitch profile according to the mean pitch and standard deviation.

Evaluation: The results from an informal listener evaluation indicate movement of the pitch contour in the proper direction; however, few of the desired individual speaker characteristics are imparted. Direct contour comparisons confirm that the new pitch contour is at a distinctly different frequency, and thus the resulting speech sentences are perceptively different from the originals. The resulting speech does not possess much of the fine prosodic structure of the desired speaker except for the proper pitch range.

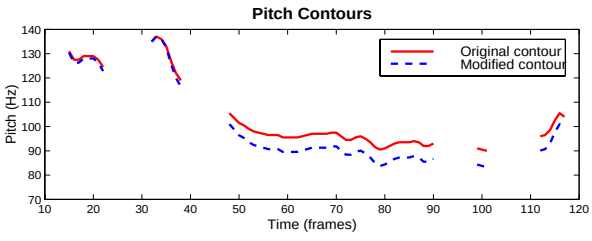

Figure 1: Example original and modified pitch contours for Gaussian algorithm

In most cases, the modified speech sounds as if the reference speaker is still producing the speech, but for some speaker pairs, listeners concluded that the speech was from the desired speaker but with some speech abnormality. This technique is not satisfactory for reproducing the desired speaker's pitch for long utterances, but it would suffice for short speech segments. Advantages to this approach are the simplicity of its implementation and the ability to use it with small amounts of training speech.

2.2. Scatterplot Pitch Modeling

This second algorithm develops a unique mapping function from the pitch of the reference speaker to that of a desired speaker. The main concepts are to generalize the Gaussian algorithm by not assuming a Gaussian distribution and to allow some level of temporal-based phoneme dependency. This algorithm begins with a training set of known pitch mappings and finds the best-fit polynomial as the selected mapping function.

Algorithm: **(A)** Begin with a database of training sentences possessing the same utterance sequence for reference and desired speakers. Estimate the mean pitch for each phone for both speakers. **(B)** Construct a scatterplot model of mean pitch for the two speakers with one data point for each voiced phone in the database. Create the data set by matching the pitch values produced by each speaker for each phone in the database. Include data only where both speakers produced the same phoneme at the same location in the utterance. **(C)** Use the method of linear least squares to find the best-fit n^{th} order polynomial for the given set of scatterplot data points. **(D)** For the input S.U.T., use the new mapping function to convert each reference pitch value to a new frequency on a frame-by-frame basis.

After experimenting with different order polynomials (1st-10th order), we chose to fit a cubic function. In practice, the algorithm sometimes yields near-linear mappings and sometimes produces functions that vary noticeably from linear. If plotted similarly, Gaussian normalization would always create a linear mapping response with control over only the slope and intercept of the line. By matching the pitch values for phones produced at the same sentence positions, it does introduce some level of context sensitivity across the sentence.

Results: Figure 2 shows a sample scatterplot of pitch values between two speakers. The solid line is the best-fit cubic polynomial for the data, and the dashed line shows the function $x = y$ for reference. Figure 3 shows the original and modified pitch contours when this algorithm is applied to the utterance, "She had your dark suit in greasy wash water all year." Note that the frequency shift is not constant but varies according to the polynomial function in Figure 2.

Evaluation: An informal listener evaluation suggests that this algorithm performs slightly better than Gaussian normalization but still imparts only a small portion of the de-

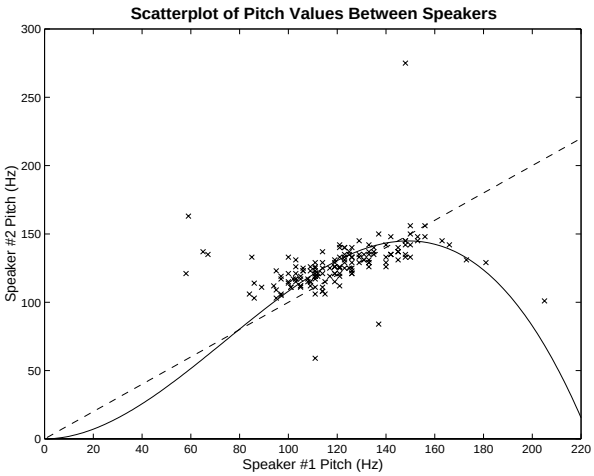

Figure 2: Example scatterplot of pitch vs. pitch

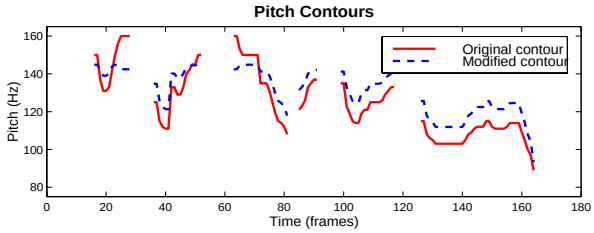

Figure 3: Example original and modified pitch contours for scatterplot algorithm

sired speaker's individual characteristics. For some speaker pairs, the results were not perceptually different than for the Gaussian algorithm. At times the resulting pitch-modified sentence was still noticeably similar to the original pitch contour, and even in those cases where the modified sentence was noticeably different from the original, the result was not necessarily perceptually closer to the desired speaker.

2.3. Sentence Contour Code Book

The code book algorithm uses DTW to select the closest pitch contour from an available training sentence database. After finding the matching pitch contour in the code book, this contour is mapped onto the S.U.T. The goal is to impart an actual sentence-level intonation contour from the desired speaker onto the S.U.T. while maintaining the existing lexical stress pattern of the original sentence. Figure 4 illustrates the flow diagram for this algorithm.

Algorithm: **(A)** Estimate the pitch contours for the input S.U.T. and for each sentence in the training database. **(B)** Find the individual time-warping path from each of the reference speaker's training sentences to the S.U.T. **(C)** Select the sentence from the database which has the smallest mismatch distance from the reference sentence as measured via DTW. **(D)** Generate a new pitch contour by warping the pitch profile closest to the S.U.T.

This algorithm uses the DTW distance to compare the S.U.T. with all sentences in the database produced by the same speaker. After finding the closest matching sentence, the algorithm selects the same sentence produced by the desired speaker. Thus, the database must contain exactly the same set of sentences produced by each speaker. Dynamic time warping is performed between the two speakers' utterances of the chosen sentence. Phone or word boundaries can be used as intermediate DTW endpoint constraints to minimize the lexical and textual effects, but this introduces the problem of aligning the phones when speakers produce different phonemes for the same utterance. The resulting warped pitch contour is time-warped again onto the sentence under test. There are not necessarily two separate warpings of the pitch profile, but the warping paths can be combined and applied in a single step.

By selecting a pitch contour from the training corpus produced by the desired speaker, this algorithm ensures that the final contour has the same pattern as that sometimes produced by the desired speaker. Using DTW to compare the same sentence from each speaker helps minimize lexical stress while maintaining the large-scale intonation differences between the speakers' production of the same sentence. Finally, warping the desired contour onto the original helps maintain aspects of the original intonation pattern while imparting characteristics of the new speaker. Since it uses actual contours from the desired speaker, this algorithm inherently moves the pitch to the proper frequency range.

Results: Figure 5 shows the original and modified pitch contours when this algorithm is applied to the utterance, "She had your dark suit in greasy wash water all year." Notice how the time-warping aligns the chosen contour as closely as possible to the original pitch contour of the sentence.

Evaluation: Compared with the other algorithms presented here, the code book algorithm showed improvements in more cases and a generally stronger approximation of the contours produced by the desired speaker. For some cases, we judged that the pitch contours shifted only slightly towards those of the desired speaker, while in other cases the pitch contours moved more noticeably. The greatest strengths of this code book algorithm are that it uses actual example contours and takes into account the comparison of known contours between the two speakers.

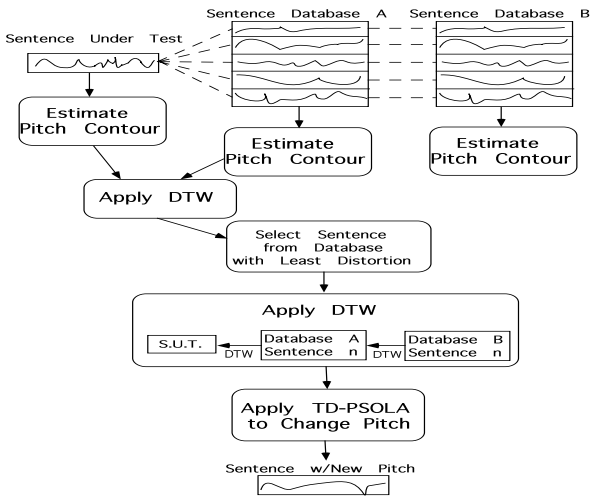

Figure 4: Flow diagram for code book algorithm

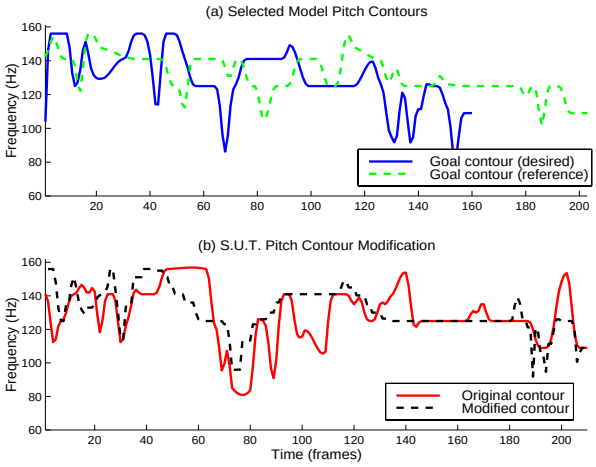

Figure 5: Example of pitch contour generation with the code book algorithm.
(a) The pitch contours chosen from the database
(b) The original and modified pitch contours

3. IMPLEMENTATION

Accurate pitch estimation is important for the present study. There are several well-known methods for pitch tracking, but we must handle the additional issue of time-warping the pitch contour for unvoiced sections of an utterance. In unvoiced speech the pitch is typically ignored, but a numeric value must be provided for DTW regardless of voicing. We achieved the best results when we generated a separate cubic spline interpolation for each unvoiced section. Moreover, when implementing these algorithms, care must be taken to prevent pitch doubling and halving effects from adversely changing the resulting models.

The code book algorithm uses dynamic time warping to determine the mapping between two different pitch contours. This study does not use a spectral coefficient scheme often used with DTW but instead directly matches the pitch super-structures. We applied Type IV local constraints as well as global constraints that limit the repetition and skip rate to two frames each. The warping is generally performed on a sentence level with endpoint matching by the limits of the voiced sections of the sentences. Where the identical sentence is available from both speakers, the warping can be improved by using additional endpoint matching to take advantage of word or phone boundaries.

Although the evaluations presented above use naturally-produced sentences, these pitch-generation algorithms are intended for use within speech synthesis systems. Thus, we have applied generated pitch contours to a concatenative speech synthesis system that uses a small database. In an informal listener test, we applied the proposed algorithms to several sentences of synthesized speech. For some sentences, the artifacts introduced by the synthesis system prevented us from being able to discern the change in pitch contour, but for other sentences the contour change was noticeable and did affect listener perception.

4. CONCLUSIONS

We have presented and evaluated several new algorithms for generating a speaker-dependent pitch contour when given an original reference utterance. The three algorithms do not attempt to reconstruct the precise pitch contour which the desired speaker would have actually produced for the sentence, but the goal is to generate contours which provide sufficient acoustic cues to persuade the listener that the desired speaker may have produced the sentence. Future TTS systems would benefit from the advantages of having such speaker-dependent contours.

The three proposed algorithms are divided into two different approaches: statistics and dynamic time warping. The statistical algorithms shift the pitch contour to the range appropriate for the desired speaker but do not change the overall shape and structure of the contour so as to impart detailed speaker characteristics. Other research [1] supports our conclusion that changing the pitch with a mapping function is not sufficient but that it is better to adjust the dynamics of the pitch structure. The time-aligning concept of DTW generally yields useful pitch contours which approach those created by the desired speaker.

Each of these algorithms carries its own advantages and disadvantages. The Gaussian algorithm is simple enough that we recommend using it as a minimum standard. The scatterplot algorithm performs better than the Gaussian algorithm, and its additional complexity and computation are usually justifiable. Pitch contours produced by the code book algorithm are of reasonable quality because they are truly representative of the contours actually generated by the desired speaker. Based on subjective evaluation, we feel that the scatterplot and code book algorithms yield the most promising results.

5. REFERENCES

[1] M. Akagi and T. Ienaga. "Speaker Individualities in Fundamental Frequency Contours and Its Control." *Proc. EuroSpeech'95*, 439–442. Sept. 1995.

[2] J. Allen, M. S. Hunnicutt, and D. Klatt. *From Text to Speech: The MITalk System*. New York: Cambridge, 1987.

[3] D. H. Klatt. "Review of Text-to-Speech Conversion for English." *J. Acoust. Soc. Am.* **82** (1987): 737–793.

[4] P. Ladefoged. *A Course in Phonetics*. 3rd ed. Philadelphia: Harcourt Brace Jovanovich, 1975.

[5] E. López-Gonzalo, J. M. Rodríguez-García, L. Hernández-Gómez, and J. M. Villar. "Automatic Prosodic Modeling for Speaker and Task Adaptation in Text-to-Speech." *Proc. ICASSP-97*, 927–930. April 1997.

[6] E. Moulines and F. Charpentier. "Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones." *Speech Communication* **9** (1990): 453–467.

[7] J. Terken and R. Collier. "The Generation of Prosodic Structure and Intonation in Speech Synthesis." *Speech Coding and Synthesis*. Ed. W. B. Kleijn and K. K. Paliwal. New York: Elsevier, 1995. 635–662.

[8] J. P. H. van Santen. "Prosodic Modeling in Text-to-Speech Synthesis." *Proc. EuroSpeech'97*, KN-19–KN-28. Sept. 1997.

[9] C. W. Wightman, S. Shattuck-Hufnagel, M. Ostendorf, and P. J. Price. "Segmental Durations in the Vicinity of Prosodic Phrase Boundaries." *J. Acoust. Soc. Am.* **91** (1992): 1707–1717.