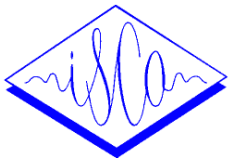

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

When will synthetic speech sound human: Role of rules and data

Jan van Santen, Michael Macon, Andrew Cronk, Paul Hosom, Alexander Kain, Vincent Pagel, and Johan Wouters

Center for Spoken Language Understanding
Oregon Graduate Institute of Science and Technology

ABSTRACT

Text-to-speech synthesis research has moved away from building general purpose systems based on an understanding of human language and speech production towards building systems based on statistical algorithms applied to large text and speech corpora, and, recently, towards building such systems for specific domains. Despite substantial progress, the overall quality of even the best systems is often still inadequate for broad user acceptance in applications that cannot also be handled with simple phrase splicing. This tutorial paper analyzes which problems must be addressed to achieve the goal of generating natural-sounding speech in limited domains in a cost-effective way, and the roles of data and rules as we work towards solutions.

1. INTRODUCTION

Current text-to-speech synthesis systems display a great variety of approaches, which is a healthy state of affairs if one believes that there is still a long way to go towards the ultimate objective of human sounding synthetic speech. These differences in approach are commonly referred to as *data* or *corpus driven* versus *rule based* approaches, but it is obvious that the differences are far from unidimensional and that they also depend on which particular speech synthesis component is being addressed.

This tutorial paper addresses the following issues:

- What is the role of *models* in the data vs. rules issue, where models are understood as mathematical formulations that attempt to either describe underlying speech production processes or capture surface regularities in data.
- What are the roles of rules, data, and models in guaranteeing acceptable system quality in a target domain?
- What does it mean for a target domain to be restricted?
- In which classes of domains does it make sense to use speech synthesis as opposed to splicing recorded words or phrases?
- What progress has been made, where do we need progress, and what directions could be explored?

2. TEXT-TO-SPEECH SYNTHESIS: BRIEF OVERVIEW

2.1. TTS Components

Speech generation has evolved over several decades [9], or even centuries [19]. Until recently, the emphasis was on generating speech from known phoneme strings, and either ignoring prosody altogether or using known pitch contours and temporal parameters; thus, this was not *text-to*-speech synthesis but merely speech synthesis.

For clarity, text-to-speech synthesis requires three logical steps:

1. *Text analysis*, which transforms symbolic input (e.g., text, tagged text, structured information stored in relational data bases) into symbolic output such as phonemes, syllables, words, phrases, or stress tags.
2. *Prosody generation*, which computes *target prosody*: timing (e.g., phoneme durations), intonation (e.g., a fundamental frequency contour sampled at 5ms intervals), and possibly contours for other acoustic features such as amplitude or spectral balance.
3. *Synthesis*, which generates speech that reflects the target phonemes and prosody.

For historical reasons, these steps will be discussed in reverse order.

2.2. Synthesis

Speech synthesis was initially performed by mechanical means, subsequently by electrical devices, and finally by digital methods (see [12] for a review). The initial digital methods were formant synthesizers, in which formant values are controlled by digital filters that have periodic input (for voiced sounds) or noise input (for voiceless sounds).

In the 1970's, a different digital method became popular, in which stored speech fragments ("acoustic units") were glued together ("concatenated") to form output speech. This development was stimulated by increased storage capabilities of computers and by the invention of speech coding techniques (in particular, linear predictive coding) that preserved naturalness at low bit rates while at the same time allowing for *prosodic signal modification* – i.e., changing the temporal and pitch characteristics of the stored speech. This development was also stimulated by what many perceived as slow progress in naturalness with formant based methods, even though in course of the years steady progress was made along these lines [1, 14, 31, 3].

Early concatenative systems used short units, typically diphones. Because a given synthetic word would have as many diphones as it has phonemes (plus one), it would have many "seams". Careless production of units would often result in these seams being audible. On the other hand, careful manual unit selection and excision can result in remarkably smooth speech [23].

Following the availability of even more storage, they idea arose that with a sufficiently large inventory of acoustic units one might avoid having seams in "risky" regions (such as rapidly moving voiced regions as opposed to "safe" regions such as voiceless closures). In addition, with even larger corpora one might perhaps also avoid making prosodic modification altogether; at least, one might not have to make drastic prosodic modifications [4]. Towards this end, instead of having a – by current standards small – computer file containing just a few hundred or thousand pre-excised acoustic units, unit choice and

excision takes place at run time using algorithms that search for an optimal sequence of speech intervals stored in a large, labeled run time speech corpus. In theory, this has the further advantage that for a given phoneme sequence several tokens may be available, from which the system could choose the one that is prosodically most appropriate and/or provides the most seamless match to neighboring units. Several of these *corpus driven* systems are now available.

2.3. Prosody generation

Both timing and fundamental frequency of natural speech are variable (the same speaker will never say the same utterance in quite the same way), and are the result of many interacting factors. Some of these factors relate directly to the communicative intent (e.g., F_0 peak height in accented syllables), others probably do not (e.g., the increase in F_0 during the first 50 ms of a vowel following a voiceless sound.) Yet, it seems risky to assume that one can model only those aspects with obvious communicative value, because one might hypothesize that for synthetic speech to sound natural all acoustic features must "come together". For example, it is well-known that formant frequencies and bandwidths, as well as spectral balance, are correlated with fundamental frequency. If one, using signal processing methods, surgically alters stored speech (as in concatenative synthesis; see below) by modifying F_0 while leaving other spectral features unchanged, the resulting speech could not physically have been produced by the speaker, and may hence sound unnatural.

For these reasons, it seems preferable to generate target temporal and fundamental frequency contours that are as similar as possible to those of natural speech.

There is a variety of ways of doing this. First, for timing the following methods have been proposed:

- Rules of the type: *If stressed, multiply duration by 1.3*
- Equations, such as the the additive model, multiplicative model, and the sum-of-products model [38].
- Classification and regression trees [28].

A shortcoming shared by these methods is they focus on overall phoneme duration, whereas it is known that factors such as phrase boundaries have non-uniform effects on the internal temporal structure of phonemes [11].

For intonation, among the standard methods are:

- Generate tonal targets, and perform interpolation between these targets [24, 25].
- Superpositional model, originated by Fujisaki [13] and later modified by Moebius and van Santen [42, 22, 43], in which the F_0 contour is decomposed into underlying phrase and accent curves.
- IPO's linear segment concatenation [33, 34].
- Festival's tilt model [10].
- Do not alter the stored speech, and use its temporal and fundamental frequency contours as is. Of course, this may create problems at seams when the natural F_0 values clash. In this case, one may still generate a target contour, but use this contour to select the right speech intervals from the corpus rather than to impose it on the units.

2.4. Text analysis

Text analysis has to perform several subtasks, ranging from relatively easy tasks (e.g., dictionary lookup) to hard tasks (e.g., determining which words are "important"; expansion of email addresses). For a more complete list:

- *Grapheme to phoneme conversion.* Among the factors making this task hard are:
  - Homographs, which may require clever algorithms that make use of contextual information (e.g., fish-related vs. music-related context in the case of *bass*).
  - Personal and geographic names, in particular in American English which has a surname type count of 1.5 million.
  - Non-words such as '%', abbreviations, acronyms, and internet addresses. As pointed out by Sproat [30], the customary approach of a non-linguistic, lookup-table or simple rewrite rule based text normalization pre-processor has become untenable. For example, the word used to pronounce the 'depends on complex contextual factors that require deeper processing than can be performed in a simple pre-processor.
- *Stress assignment.* Determining which syllable in a word may receive stress is reasonably easy in many languages.
- *Pitch accent or emphasis level/type assignment.* Pitch accents and emphasis often reflect complex semantic relations that are generally beyond the reach of current systems.
- *Phrasing.* Phrasing is another hard problem, partly due to semantic information needing to be taken into account.

In general, the difficulty of the various subtasks vary with the nature of the input materials and the language. For example, children's books can be expected to be simple in terms of grapheme to phoneme conversion and phrasing. However, accent assignment must reflect a solid understanding of the semantics by the system. On the other hand, reverse directory assistance poses mostly problems in terms of the name pronunciation subtask.

Given the variety of the challenges posed by these subtasks, it is unlikely that a single methodology can be developed that works optimally in each case. It is also unlikely that even for a given subtask, the same approach is optimal independently of the target domain.

3. SYNTHESIS QUALITY

The second key issue mentioned in the Introduction section – the roles of rules, data, and models in guaranteeing acceptable system quality in a target domain – assumes that there are agreed upon ways for measuring synthesis quality. It should be obvious that TTS performance – regardless of how it is measured – is highly sensitive to the characteristics of the input text, the application in which the system is embedded, the hardware platform, the acoustics of the listening context, and the users of the system [35, 26, 27].

At trade shows, often the vendor determines the text, which makes it unlikely that the text was selected to highlight serious deficiencies in the system. If a customer can type in the text, this also constitutes a seriously flawed evaluation because the customer knows the text, and thus cannot assess the system's intelligibility.

Although trade shows are obviously focused on convincing the customer to buy products and are therefore fundamentally biased (but surprisingly effective), performing a useful evaluation is harder than one thinks. The key for a good evaluation is that one thoroughly understands the characteristics of the situation in which the system is intended to be used (the target situation), and that one gathers sufficient information about the TTS system to predict the system's performance in the target situation.

Nevertheless, in this paper will assume that the quality differences being discussed are sufficiently salient that most ways of measuring quality would be in agreement.

4. DOMAINS

Until recently, most TTS systems were not focused on any particular domain. At the other extreme, word and phrase splicing systems were entirely focused on one specific application.

It has become clear that much can be gained by imposing domain limitations. But the challenge is that if one defines a domain sufficiently narrowly that it can be covered with a tolerable effort by word and phrase splicing systems, the purpose of using TTS is defeated.

But first, we have to understand better what properties of domains matter, and the role of *coverage*.

System construction involves various uses of text and speech corpora, e.g.:

- A pronunciation dictionary used for training and testing a machine learning algorithm.
- A speech corpus from which labeled acoustic units are excised (standard concatenative synthesis).
- A speech corpus that is segmented and labeled for runtime acoustic unit selection (corpus driven concatenative synthesis).
- A segmented corpus used for estimating parameters of a duration model.
- A segmented corpus used for classification and regression tree training.
- Perceptual and other tests of the system on a corpus of test text items.

Once a system has been constructed and tested, the key questions are:

- What is the target domain?
- What is the relationship between the target domain and the corpora used for system training and testing?
- What is our basis for the assertion that we can *generalize* from system performance on the test corpora to system performance on the target domain?

4.1. Two dimensions of target domain perplexity

Abstractly, a domain consists of a set of text items. This set is in practical terms infinite in the case of unrestricted domain TTS systems, and is small enough to be recorded verbatim in the case of word and phrase splicing systems. Roughly speaking, domains vary on two dimensions: vocabulary size and the variety of sentence structures. Both dimensions directly affect the *perplexity* of a domain (basically, how many different words might follow a particular word), but we prefer to keep these dimensions separate because they differ in their effects on individual TTS components. For example, the domain of arbitrary digit sequences (including social security numbers, and ZIP codes) has a small vocabulary that can be easily covered by a set acoustic units that are relatively long (e.g., the *th-r-iy-f* unit used in the digit sequence 3-5) and are easy to concatenate (*th* and *f* are good cutpoints). On the other hand, generating good prosody for arbitrary digit sequences is less easy, because one needs to address phrasing (e.g., "10003" $\rightarrow$ "1,0003" but perhaps "97006: $\rightarrow$ "97,006") and accenting (it certainly sounds awkward if all digits in a digit sequence have the same type and degree of accentuation). On the other hand, the domain consisting of sentences of the type "You have a call from <*first name*> <*last name*>" does not pose great challenges for the phrasing and accenting components; but if, as in the U.S., the first name can be any of thousands and the last name can be any of 1.5 million, then this poses a serious challenge to the text-to-phoneme transcription and the signal processing components.

4.2. Target domain frequency distributions

The same domain has different sizes depending on which TTS component one is referring to. One way to define these sizes is in terms of the set of feature vectors that a given component has to address. For example, the feature vectors that form the input to a phrasing component may consist of parts of speech tags in window of five words, whereas the feature vectors for a timing component may consist of phoneme identities, stress, emphasis, and position.

What seems to be a general rule with language materials is that the probability of *some* very infrequent vector occurring in a small text sample is very large. In other words, *rare things happen frequently*. Of course, this does not pose a problem if the total number of feature vectors is small enough that we can test the system on all of them. But it does pose a serious problem if the set is large, and we do not have independent reasons for believing that component output quality will generalize well outside of the test set. As we shall see below in Section 5., there are situations where we might have good reasons for confidence in a component's generalization capabilities, and other situations where this confidence is unjustified.

A second notable fact about these feature vectors is that the frequency distributions of these vectors vary considerably between different domains [40]. This means that, again unless one has strong independent reasons for believing in the generalization capabilities of a system beyond the test materials, generalizing from one textual genre to another is particularly hazardous. To give a concrete example: Among the first large text corpora becoming available on-line on a large scale were mostly newspaper corpora. As a result, several TTS systems have been trained and constructed on the basis of text that, while certainly not too restricted, nevertheless may generalize poorly to domains encountered in a variety of niche applications, such as drug names,

airport names, digit sequences, and stock tables. Yet, it is precisely in such niche applications that speech synthesis is increasingly more put to commercial usage.

4.3. Generalization and component structure

Confidence in generalizing from test and training materials to a target domain should depend on what the component learns or represents. For example, consider a system's capability to pronounce names. In the U.S., it us quite difficult to produce high-accuracy rule based systems. Instead, mixtures of dictionaries and rules are used, at times in a framework where first a name is classified in terms of ethnic origin and then subject to origin-specific rules or dictionaries. These dictionaries are rarely exhaustive, for the simple reason that the number of names is quite large. Usually, dictionaries only contain entries for the most frequently occurring items. But that means that, unless the "fall back" rules used for items not in the dictionary are extraordinarily clever, there is no basis to assume that the system as a whole will perform adequately on less frequent names, in particular because the less frequent names are more than likely to involve small ethnic groups whose languages of origin differ strongly from the dominant languages. We use the term "list-like" component for components such as this.

By contrast, formant synthesizers consist of a small set of principled quantitative rules that can be tested fairly exhaustively. We will revisit component structure below in Section 5.

4.4. Summary

In summary, generalizing from adequate system performance on the test domain to adequate system performance in the target domain take several leaps of faith. The coverage of the target domain by the test domain has to be analyzed at several levels in terms of the feature vectors that form the input to the system components. Throughout, one has to be aware of the large differences in feature vector distributions between different domains.

5. RULES, DATA, AND MODELS

To repeat, the key issue in Speech synthesis is how we can confidently generalize from system performance on the test corpora to system performance in the target domain. We will analyze the merits of rule-, model-, and data- based approaches solely on the basis of this criterion. Thus, we shall assume that the system provides adequate quality on the test and training data, and that the only remaining issue is whether we can promise equally adequate performance in the target domain. We add the stipulation that the target domain is non-trivial, by which we mean that it cannot be covered by simple phrase splicing at an equal or lesser cost. As an aside, it should be noted that many speech technology companies are willing to spend surprisingly large resources on making recordings for phrase splicing – as if they are trying to avoid speech synthesis at almost any cost, which poses a challenge to those who maintain that speech synthesis is a "solved problem."

Although we do not intend for this paper to be a philosophical essay, we have to clarify how we will use the terms *rule*, *model*, and *data*. Our usage, of course, may differ from that of others.

We assume that the term *data* does not really have to be defined. We just point out that during system construction one rarely deals with "raw" data. For example, one typically deals with representations in terms of quantitative measures (cepstral parameter vectors, durations) or linguistic labels. As any philosopher of science would point out, the choice of a measurement or labeling scheme hides countless theoretical assumptions; but we shall not address these here.

5.1. Rules

For historical reasons, the term "rule" mostly reflects the contributions of the linguistic community to speech synthesis, and is therefore reserved for text analysis and, to some degree, prosody generation. For the purposes of this discussion, we shall assume that rules are generated manually, based on linguistic knowledge. A typical example of a rule is:

/T/ preceded and followed by stressed vowel $\rightarrow$ /DX/

Far more intricate rules exist, such as those involving morphological decomposition [30]. Sproat provides examples that are unlikely to be learned from a reasonably sized training corpus by machine learning methods, such as the many pronunciations of the "%" sign in Russian discussed above.

Manual text analysis rule generation has the following potential drawbacks:

- It is labor intensive and requires special expertise
- It may result in rule systems whose overall behavior and coverage is hard to assess.
- *Sequential* rule systems, where decisions in early rules or rule modules cannot be reversed by subsequent rules or modules, are inherently fragile. However, rules represented as (weighted) finite state transducers do not have this irreversibility drawback.

Strong claims have been made about the success of machine learning techniques for building spelling-to-sound components [36]. The overall correct pronunciation rates of 90-95% (per phoneme, on test data) that have been reported, although impressive, are not adequate for high-quality speech synthesis, because these per-phoneme rates translate into per-word rates of less than 80%. High-end commercial TTS systems have much higher rates, perhaps as high as 95%. In practical terms, the question is not how easy it is to attain this performance level, but what resources are required to attain an acceptable accuracy level, e.g. 95%.

It should also be noted that once a system has been built for a particular language or dialect, the text analysis component does not need to be changed substantially if at all when we want to change the voice of the system. The latter involves changes in the speech corpus, and possibly changes in the prosody generation parameters; but rarely changes in the spelling-to-sound component. Perhaps construction of the spelling-to-sound component is least in need of being automated. (Below, however, we shall claim that these techniques may be quite useful for rapid development of spelling-to-sound components for homogeneous subdomains.)

There is a growing consensus that a complete spelling-to-sound system is constructed most efficiently by supervised learning algorithms that are initialized with manually generated rules

that are easy to state and have a high level of validity. Most languages have been subjected to extensive linguistic analyses, resulting in many such rules; it would be silly not to make use of that. The algorithms would generate new rules for manual inspection, find instructive test cases for a proposed rule, and estimate parameters such as the weights in weighted finite state transducer based systems.

5.2. Models

The term ”model” is reserved for quantitative assertions, at the prosody generation and synthesis levels. Also, the term carries the connotation of representing aspects of the structures or processes involved in speech production. For example, the linear predictive coding scheme has an interpretation in terms of a speech production model according to which the vocal chords function like an excitation source operating independently of the vocal tract, and where the vocal tract can be described by a constellation of cross-sectionally round, rigid pipe fragments. Likewise, the Fujisaki intonation model is based on specific assumptions about vocal chords.

The multiplicative model used for segmental duration prediction [37] does not characterize underlying processes of structures as obviously as the LPC and Fujisaki models. However, it rests on broad assumptions about speech production, such as: speakers increase duration for stressed sounds; allophones require specific articulatory patterns that are responsible for one allophone to always have a longer duration than another allophone, holding all else constant.

The defining characteristics of these models and how they are used in TTS can be summarized as follows:

- Their *structure* (i.e., the mathematical formalism without specific parameter values) is based on assumptions about the speech production process. We expect these assumptions to be valid for the target domain, or, in many instances, for human speech in general. For example, we do not expect speakers to shorten stressed vowels in some textual domain.
- Training data are used to estimate parameters. These parameter estimates may or may not be valid for the target domain. For example, if both the word ”to” and the digit ”two” are transcribed the same (which they shouldn’t), then duration predictions for the /U/ will be much too short for the word ”two” if they are based on a corpus with a preponderance of instances of ”to”.

We now consider usage of general purpose prediction engines, such as classification and regression trees (CART) as applied to segmental duration prediction [28]. In a typical application, the training corpus consists of a table with feature vectors and observed durations. The end result of training consists of a tree, where each node is tagged by the name of a factor (e.g., ”phrasal location”) and a dichotomy of values on these factors (e.g., $\{\{\text{final}\}, \{\text{initial, medial}\}\}$). The structure of the tree may vary from one training sample to the next. This structure is not *structural* in the same sense as the Fujisaki model’s formalism or the multiplicative model are structural. In other words, CART tree structure is a parameter, be it not a continuous quantitative parameter. There is, in fact, very little that is structural about CART. This is the other side of the major attraction of CART, which is that it can be used as a general purpose technique in a wide variety of applications.

These assertions are nicely illustrated by Maghbouleh [21] who found that a knowledge based approach [39] generalized much better than CART to test materials drawn from a different corpus than the training materials. In fact, even when CART was given two orders of magnitude more training data than the knowledge based approach, the difference in performance hardly decreased.

In general: The more assumptions that can be made that are valid for the target domain (and preferably for human speech in general), and are built into the model, the more confident we can be that the model’s predictions will generalize to the target domain. This is particularly true when the training corpus has poor coverage of the target domain, by being small or being outright different. If, on the other hand, the training corpus covers most feature vector *types* that can occur in the target domain, then the difference in generalization power between general purpose techniques and model based approaches becomes smaller.

5.3. Corpus driven synthesis

How does corpus driven synthesis fit in this conceptual framework? Are there any rules and models, perhaps implicitly?

To reiterate, a corpus driven synthesis system characterized by the following:

- Three *cost functions*, that, independently of each other, can be *quantitative* (e.g., some distance measure between actual and target F_0) or *symbolic* (goodness of the match in terms of a symbolic representation of phonemes and prosodic tags.).
  - *Concatenation cost*, measuring the discrepancy between excised units.
  - *Intrinsic quality cost*, measuring the ”goodness” of a phonetic segment when not used as a splice point.
  - *Prosodic cost*, measuring the prosodic appropriateness of a potential unit given the prosodic target
- Concatenation procedure, which *optionally* involves:
  - Spectral smoothing at splice points
  - Prosodic modification to bring the stored speech in line with the target prosody.

The Laureate system [5] is unique among corpus driven systems in that it only uses symbolic costs. Of course, the symbolic representation used is quite rich, and includes characteristics of contextual segments that will considerably reduce the acoustic variability, and thereby any quantitative costs if these were measured. In other systems, such as the CHATR system [4], costs are quantitative.

Corpus based synthesis system performance in target domains rests on two factors:

- The *perceptual validity* of the cost measures.
- The *coverage* of the target domain by the training corpus.

A few remarks about cost measures. Recent studies [17, 45] have shown that cost measures based on distance measures do

not predict perceptual quality accurately, although some measures are definitely superior to others. Some informal experiments by the first author (with Albert Febrer of the Universitat Politecnica de Catalunya) suggested that listeners are surprisingly indifferent to large acoustic distances, e.g. as measured in formant space. In summary, we are currently still ignorant about how to measure acoustic costs.

However, there are scenarios where acoustic cost measures can be valid, basically because they are trivial. For example, the acoustic cost measure that assigns a value of 0 to two acoustic units whenever their splice point involves a voiceless stop closure or a pause, and 1 otherwise, will almost certainly produce smooth speech. The catch is, of course, that the odds that arbitrary input sentences can be generated using only such units is astronomically small, unless the training corpus is quite large relative to the domain [41].

A further complication is created for systems that do not perform prosodic modification of the stored speech. First, it makes it that much harder to find acoustic units because these units must have low prosodic costs in addition to low concatenation and intrinsic quality costs. Second, we are not aware of any studies measuring the perceptual validity of prosodic cost measures. This is likely to be complicated because of the importance of temporal factors. It has been shown that slight changes in alignment of the F_0 contour with the segmental boundaries can change intonational meaning [18, 8, 2]. This means that simple measures based on the frame-by-frame differences between target and stored original F_0 contour are probably inadequate. Of course, if these differences are vanishingly small we can be assured that the target contour is faithfully matched by the stored contour. But if the differences are not vanishingly small, then, similarly to the other costs measures, we basically have no idea what the measured distance reflects in perceptual terms.

6. SOME PROPOSALS

We discuss here some possible solutions that we consider promising.

6.1. Text analysis

Input text is often heterogeneous in the sense that it contains distinct regions. For example, an email message contains a header, a signature block, embedded messages, or tables. And regular newspaper text contains headings, tables, acronyms, and digit sequences. We propose an approach in which input would first be classified into distinct regions and then is processed in parallel. This approach contrasts with the more traditional approaches that have great difficulty parsing input such as "His email address is dd@algorithm.com", because they attempt to apply the same algorithms as are applied to regular text. For example, the dot may be inferred to be an end of sentence. We claim that a better approach would be to first recognize email addresses and than apply analyses optimized for email addresses.

Another example would be a table. Again, raw ascii tables tend to be incomprehensible when processed by a standard TTS system. However, if we have a table recognizer then the problem of how to speechify a table becomes much easier.

In addition, we think that it is plausible that machine learning algorithms may have an easier time with homogeneous text than with text that is an amalgam of different types.

6.2. Prosody generation

Preserving micro-details and *modifying natural F_0 contours.* Natural F_0 contours are not smooth at a microscopic level. That is, we may observe a single-peaked rise-fall-rise pattern that looks smooth, the first derivative of this pattern is far from smooth. An obvious example if formed by *creaking*, in which pitch periods double in length. Such irregularities pose a serious challenge for prosodic signal modification techniques. Of course, one could eliminate creaky regions from the training corpus, but creaking is in fact an important prosodic cue that would enhance perceived naturalness.

The problems faced by prosodic modification techniques would be lessened if it were possible to decompose the natural F_0 contour into an underlying smooth contour and a "residuals" contour that contains the local irregularities. During synthesis, the target contour would replace the smooth natural contour, and the same time warping operations would be applied to the spectral representation and the residuals contour, thereby guaranteeing that the latter two are kept in sync.

Beyond F_0 and timing (i) There is substantial evidence that prosody involves more than timing and F_0 [29], and also involves spectral balance or tilt, and formant values [6]. This means that the prosody module has to also provide target contours for these features.

6.3. Synthesis

Prosodic signal modifications: F_0, timing, and more Current waveform-based synthesis methods do not have have problems making relatively small modifications of timing and F_0 in recorded segments, by using techniques like PSOLA [7] or sinusoidal modeling [20, 32]. The reason these techniques work well is that they are able to modify signals along two independent "perceptual dimensions" (time duration and fundamental frequency), while leaving almost all other detailed characteristics of the signal intact. This is in stark contrast to techniques like formant synthesis or pulse-excited LPC, which first assume an extremely constrained model of the speech signal as a whole, having many fewer parameters than the speech itself (i.e., a small number of resonances excited by a periodic pulse train). The "synthetic" character of formant- or LPC-coded speech arises because there are fine, quasi-random details of the signal that are not described by the model. In contrast, the waveform modification technique assumes a much less constraining model - essentially only that speech consists of a set of pulses that are spaced at multiples of $1/F_0$ from each other and vary slowly from one to the next. The "parameters" of this model are the waveform samples themselves, but critically, there is still some degree of *control* of the perceptually-relevant dimensions of the signal.

We believe there is much to be gained by extending this paradigm beyond small adjustments of F_0 and duration, to allow for modification of broad spectral characteristics as well. For example, a powerful extension would be a technique that allows formant trajectories in recorded speech to be altered, while still maintaining other detailed characteristics (e.g., the breathiness of a soft female voice). In [46], we propose a spectral modification technique that can accomplish very high-quality modification of spectral shape, by describing the difference between original and target spectra by a combination of a frequency warp

and gain equalization, and preserving spectral detail while modifying broad-scale characteristics. By using this technique as a platform, we hope to allow many of the kinds of control strategies long-used in formant synthesis (e.g., modeling vowel reduction by a formant "target undershoot" model [Lindblom63]) to carry over into the realm of concatenative synthesis. This can potentially allow concatenative databases to be much smaller, since vowel reduction and other allophonic variations could be modeled as transformations of other units, instead of being exhaustively recorded. A second application of this approach is to represent the interaction of modifications of $F0$ with spectral envelope (vocal tract) changes [16].

By modeling certain acoustic effects as *transformations* of recorded speech data, we can more efficiently cover the space of sounds that humans make in producing speech, without assuming an overly-simplistic model.

6.4. Tools

Automatic segmentation and labeling One of the challenges of corpus based approaches is that they require a large corpus to be segmented and labeled. This puts a premium on the development of high-accuracy automated methods. Recently, significant progress has been made with automatic segmentation [15, 44], and we would not be surprised if certain types of speech materials can be processed entirely automatically in the near future.

Acoustic cost As discussed earlier, acoustic cost measures are critical for the success of corpus based methods, because the acoustic units are too large in number for any type of off-line inspection or quality control. We propose that systematic perceptual experiments are required to find out which distance measures are optimal. More specifically, we suggest that parameterized families of measures are used that also incorporate the dynamics of the speech signal, and that different measures may be required for different classes of phonemes.

7. CONCLUSION

Speech synthesis is not a solved problem. In fact, as we observe the historical development that started with brave attempts to directly model human speech generation and that now has reached a point that is remarkably close to simple word and phrase splicing, one may wonder what the nature of the progress has been. Yet, there is no doubt that the best synthesis systems sound much better than 10 years ago, in particular those systems that apply corpus based approaches to narrow domains.

The key challenge faces by the latter group of systems is cost effectiveness – can one build such a system with fewer resources than a word and phrase splicing based system at a comparable quality level?

We conjecture that the answer to the latter question is not certain, and is probably mixed – affirmative for certain domains and negative for others.

The general opinion expressed in this paper is that over the years many useful tools and insights have been produced. But because they have been produced in the contexts of different traditions and disciplines, not enough attention has been paid to eclectic approaches that make use of any good tool, regardless of its philosophical or historical baggage.

At the same time, obviously far more tools and insights are needed. Most importantly, we believe that a key challenge lies in designing architectures that combine these assets, and that allow data, models, and rules to play the roles they are best suited for.

REFERENCES

[1] Allen, J., Hunnicut, S., and Klatt, D. *From text to speech: The MITalk System*. Cambridge University Press, Cambridge, U.K., 1987.

[2] Arvaniti, A., Ladd, D., and Mennen, I. Stability of tonal alignment: the case of Greek prenuclear accents. *Journal of Phonetics 26* 1998, 3–25.

[3] Bickley, C. A., Stevens, K., and Williams, D. A framework for synthesis of segments based on pseudoarticulatory parameters. In *Progress in speech synthesis*, J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, Eds. Springer, 1996, pp. 211–220.

[4] Black, A., and Taylor, P. CHATR: a generic speech synthesis system. In *Proc. COLING94* (1994), pp. 983–986.

[5] Breen, A., and Jackson, P. Non-uniform unit selection and the similarity metric within BT's Laureate TTS system. In *Third ESCAWorkshop on speech synthesis* (Jenolan Caves, Australia, 1998).

[6] Caspers, J. *Pitch movements under time pressure*. PhD thesis, Leiden University, 1994.

[7] Charpentier, F., and Moulines, E. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. In *Proc. of Eurospeech-89* (Paris, 1989), pp. 13–19.

[8] d'Imperio, M., and House, D. Perception of questions and statements in Neapolitan Italian. In *Proceedings of the Fifth European Conference on Speech Communication and Technology* (Rhodes, September 1997).

[9] Dudley, H. The vocoder. *Bell Labs Rec. 17* 1939, 122–126.

[10] Dusterhoff, K., and Black, A. Generating f0 contours for speech synthesis using the Tilt intonation theory. In *Proceedings ESCA Tutorial and Research Workshop on Intonation: Theory, Models and Applications* (Athens, September 1997).

[11] Edwards, J., and Beckman, M. E. Articulatory timing and the prosodic interpretation of syllable duration. *Phonetica 45* 1988, 156–174.

[12] Flanagan, J. L. *Speech analysis, synthesis and perception*, vol. 3 of *Kommunikation und Kybernetik in Einzeldarstellungen*. Springer, Berlin, 1972. 2. erw. Aufl.; 1. Aufl. 1965.

[13] Fujisaki, H. Dynamic characteristics of voice fundamental frequency in speech and singing. In *The production of speech*, P. F. MacNeilage, Ed. Springer, New York, 1983, pp. 39–55.

[14] Hertz, S. The Delta programming language: an integrated approach to nonlinear phonology, phonetics, and speech synthesis. In *Papers in Laboratory Phonology I: Between the Grammar and Physics of Speech*, K. J. and M. E. B. M.E., Eds. Cambridge: Cambridge University Press, 1990, pp. 215–257.

[15] Hosom, J.-P. *Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information*. PhD thesis, Oregon Graduate Institute, 2000.

[16] Kain, A., and Stylianou, Y. Stochastic modeling of spectral adjustment for high quality pitch modification. In *Proceedings of IEEE ICASPP 2000* (2000).

[17] Klabbers, E., and Veldhuis, R. On the reduction of concatenation artifacts in diphone synthesis. In *Proceedings ICSLP* (Sydney, Australia, 1998).

[18] Kohler, K. Macro and micro F0 in the synthesis of intonation. In *Papers in Laboratory Phonology I: Between the Grammar and Physics of Speech*, J. Kingston and M. Beckman, Eds. Cambridge: Cambridge University Press, 1990, pp. 115–138.

[19] Kratzenstein, C. G. Sur la naissance de la formation des voyelles. *Journal de Physique 21* 1782, 358–380. French translation of: Tentamen coronatum de voce, Acta Acad. Petrop., 1780.

[20] Macon, M. W. *Speech synthesis based on sinusoidal modeling*. PhD thesis, Georgia Tech., October 1996.

[21] Maghbouleh, A. An empirical comparison of automatic decision tree and hand-configured linear models for vowel durations. In *Proceedings of the Second Meeting of the ACL Special Interest Group in Computational Phonology* (1996), Association for Computational Linguistics.

[22] Möbius, B. Components of a quantitative model of German intonation. In *Proceedings of the 13th International Congress of Phonetic Sciences* (Stockholm, 1995), vol. 2, pp. 108–115.

[23] Olive, J. A new algorithm for a concatenative speech synthesis system using an augmented acoustic inventory of speech sounds. In *Workshop on speech synthesis* (Autrans France, 1990), ESCA, pp. 25–30.

[24] Pierrehumbert, J. *The Phonology and Phonetics of English Intonation*. PhD thesis, Massachusetts Institute of Technology, September 1980. Distributed by the Indiana University Linguistics Club.

[25] Pierrehumbert, J. Synthesising intonation. *Journal of the Acoustical Society of America 70*, 4 1981, 985–995.

[26] Pols, L., van Santen, J., Abe, M., Black, A., House, D., Liberman, M., and Wu, Z. Easy access via a tts website to mono- and multilingual text-to-speech systems. In *Proceedings Elsnet in Wonderland* (Soesterberg, The Netherlands, 1998).

[27] Pols, L., van Santen, J., Abe, M., Kahn, D., and Keller, E. The use of large text corpora for evaluating text-to-speech systems. In *Proceedings First International Conference on Language Resources and Evaluation* (Granada, Spain, May 28–30 1998), vol. 1, pp. 637–640.

[28] Riley, M. Tree-based modeling for speech synthesis. In *Talking machines: Theories, models, and designs*, G. Bailly and C. Benoit, Eds. Elsevier, 1992, pp. 265–273.

[29] Sluijter, A. *Phonetic correlates of stress and accent*. Holland Institute of Generative Linguistics, 1995.

[30] Sproat, R., Möbius, B., Maeda, K., and Tzoukermann, E. Multilingual text analysis. In *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*, R. Sproat, Ed. Kluwer, Boston, MA, 1997, ch. 3, pp. 31–87.

[31] Stevens, K., and Bickley, C. Constraints among parameters simplify control of Klatt formant synthesizer. *Journal of Phonetics 19* 1991, 161–174.

[32] Stylianou, Y. *Harmonic plus noise models for speech, combined with stat. methods for speech and speaker modification*. PhD thesis, ENST, January 1996.

[33] 't Hart, J., Collier, R., and Cohen, A. *A Perceptual Study of Intonation*. Cambridge University Press, Cambridge UK, 1990.

[34] Terken, J. Synthesizing natural-sounding intonation for Dutch: rules and perceptual evaluation. *Computer Speech and Language 7* 1993, 27–48.

[35] van Bezooijen, R., and van Heuven, V. Assessment of Synthesis Systems. In *Handbook of standards and resources for spoken language systems*, D. Gibbon, R. Moore, and R. Winsky, Eds. Walter de Gruyter & Co, Berlin, 1998, ch. 12, pp. 481–563.

[36] Van den Bosch, A. *Learning to pronounce written words. A study in inductive language learning*. PhD thesis, Universiteit Maastricht, 1997.

[37] van Santen, J. Contextual effects on vowel duration. *Speech Communication 11* 1992, 513–546.

[38] van Santen, J. Analyzing N-way tables with sums-of-products models. *Journal of Mathematical Psychology 37*, 3 1993, 327–371.

[39] van Santen, J. Assignment of segmental duration in text-to-speech synthesis. *Computer Speech and Language 8* April 1994, 95–128.

[40] van Santen, J. Combinatorial issues in text-to-speech synthesis. In *Proceedings Eurospeech-97* (Rhodos, Greece, 1997).

[41] van Santen, J. Combinatorial issues in text-to-speech synthesis. In *Proceedings of Eurospeech-97* (Rhodes, September 1997).

[42] van Santen, J., and Hirschberg, J. Segmental effects on timing and height of pitch contours. In *Proceedings ICSLP '94* (1994), pp. 719–722.

[43] van Santen, J., and Möbius, B. Modeling pitch accent curves. In *Proceedings ESCA Tutorial and Research Workshop on Intonation: Theory, Models and Applications* (Athens, September 1997).

[44] van Santen, J., and Sproat, R. High-accuracy automatic segmentation. In *Proceedings of Eurospeech-99* (Budapest, Hungary, September 1999).

[45] Wouters, J., and Macon, M. A perceptual evaluation of distance measures for concatenative synthesis. In *Proceedings ICSLP* (Sydney, Australia, 1998).

[46] Wouters, J., and Macon, M. Spectral modification for concatenative speech synthesis. In *Proceedings of IEEE ICASPP 2000* (2000).