# Shona Processor and Synthesizer that Converts Speech to Text and Text To Speech

**Muzheri Kernan**
Department of Computer Science,
National University of Science and Technology,
Zimbabwe
kmuzheri@nust.ac.zw

**Chilumani Khesani Richard**
Department of Computer Science,
National University of Science and Technology,
Zimbabwe
khesani.richard.chilumani@nust.ac.zw

**Abstract.** *Natural language processing is a challenging area that has been of interest to researchers in fields like Human Computer Interaction, Artificial Intelligence and Intelligent Data Processing/ Simulation. The complexity of computationally encoding and processing speech data from diverse voices and languages requires robust heuristics rules for sound signal processing. We attempt to process Shona, one of Zimbabwe's official languages by concatenation and synthesis based on Hidden Markov Models. This implies transforming and smoothly connecting captured voice samples from a target Shona speaker. In this paper we explain and discuss a spectral processing approach used to build a working prototype that transforms the Shona speech samples with the best quality and the highest flexibility possible. We also discuss how the prototype uses spectral speech models that are aware and take advantage of the processes involved in voice production.*

**Keywords:** shona, speech, text, voice

## INTRODUCTION

Voice processing and synthesis is a series of transformations that are applied to a voice data signal when it has been captured. These transformations may include amplification, phase transformation, saving to a storage device, voice alterations, conversion from spoken words to text, conversion from text to sound imitating natural voice, etc. While voice recognition may be considered under voice processing, the term *voice recognition* is generally used to refer to biometric recognition systems that must be trained to a particular speaker—as is the case for most desktop voice recognition software.*Speech recognition*, is part of voice data processing, a solution which refers to technology that can recognise speech without being targeted at a single speaker such as a call centre system that can recognise arbitrary voices (Junqua and Haton, 2005). Two major limitations of voice processing and synthesis have been the extensive amount of time required by the user and/or system provider to train the software, and the inaccuracy of the underlying models used to represent voice data. Thus voice processing and synthesis has not yet realised its full market potential (Davies, et. al, 2002). The performance of speech recognition systems is generally specified and/or measured in terms of accuracy and speed. Accuracy is rated with *word error rate* (WER), whereas speed is measured with the *real time factor*.

In this paper we discuss methods for *speech-to-text* and *text-to-speech* conversion based on Hidden Markov Models. This approach has been chosen as a way to improve the quality of the conversion. Thus, it would be a desirable dream for many Shona speakers, translators and converters to have tools capable of synthesising speech or real sounding voices any-time and anywhere and producing the relevant text with minimal error possible. We share this dream and believe that this reseach contributes one more step towards this challenging goal.

# REVIEW OF RELEVANT TECHNIQUES AND TECHNOLOGIES

Substantial research on natural language processing has been done by both published and unpublished researchers. Several decades has been devoted to study the natural acoustical properties of speech and voice to understand the details of the mechanisms involved in voice production. With the appearance of sound synthesis techniques, special emphasis has been devoted to imitate the processes involved in speech voice production and to find ways to reproduce them and/or convert them to text by means of signal processing techniques (NIST, 2009). Among the many existing approaches to voice processing and synthesis of speech sounds, the ones that have had the most success are without any doubt the sampling based ones, which sequentially concatenate samples from a corpus database.

The success of sampling relies on the simplicity of the approach. It just samples existing sounds, but most importantly it succeeds in capturing the naturalness of the sounds, since the samples are real sounds. However, sound synthesis is far from being a solved problem and sampling is far from being an ideal approach. The lack of flexibility and expressivity are two of the main problems, and there are still many issues to be worked on if we want to reach the level of quality comparable to that of professional speakers, translators and converters. Sampling based techniques have been used to reproduce practically all types of sounds. Probably the speech voice has been synthesised with the least success in the sound synthesis field, despite the many efforts devoted to achieve a realistic, natural sounding, and expressive result. This has left *text-to-speech* and *speech-to-text* conversion with so much unsatisfactory results. In subsequent subsections we discuss the different techniques and technologies for *text-to-speech* and *speech-to-text* conversions.

## Language Modelling

Language Models (LMs) capture regularities in spoken language and are used in speech recognition to estimate the probability of word sequences. While grammatical constraints described by hand-crafted context-free grammars have been used for small to medium size vocabulary tasks, Large Vocabulary Continuous Speech Processing (LVCSP) is essentially always based on data driven approaches (Davis, 2000). The statistical method we use for language modelling is the n-gram model, which attempts to capture the syntactic and semantic constraints of the language by estimating the frequencies of sequences of n words. The assumption is made that the probability P(W) of a given word string W=($w_1$, $w_2$,... $w_k$) can be approximated by the following forward sequential decomposition.

$P(W) = \prod^{\mathbf{k}}_{\mathbf{i=1}} \{Pr(w_i \,|\, w_{i-n+1}, ..., w_{i-2}, w_{i-1})\}$

thereby reducing word history to the preceding n – 1 words. It should be noted that decompositions of P(W) can also be appropriate, for example, a backward decomposition leads to a backward n-gram model. A prerequisite for estimating n-gram language models is the availability of appropriately processed text corpora. Language models are estimated from manual transactions of speech corpora and from normalised text corpora. To ensure accurate models, the texts need to be as representative as possible of the expected audio input to be transcribed. Text preparation entails locating appropriate sources of text data and audio transcriptions, and processing them in a homogeneous manner. Language models are generally optimised and compared by measuring the perplexity of a set of left out data, referred to as LM development data. This test set perplexity of the language model M is depends on both the language being modelled and the model, i.e., it gives a combined estimate of how good the model is and how complex the language is (Rabiner, 1989). If the left out data set is representative of the model, the perplexity can be seen as a measure of the average branching factor, i.e., the vocabulary size of a memoryless uniform language model with same entropy as the language model under consideration.

$$\mathrm{Px}(T|M) = P(T|M)^{-\frac{1}{L}} \simeq \left(\prod_{i=1}^{L} P(w_i|w_{i-2}, w_{i-1})\right)^{-\frac{1}{L}}$$

## Text Preparation

Although ideal language model training data would consist of large corpora of transcribed audio data representative of the targeted task, in practice such data are difficult to obtain. Therefore, a variety of other more or less closely related text materials are used for language model training. Given a large text corpus it may seem relatively straightforward to construct n-gram language models. Most of the steps are standard and make use of tools that count word sequence occurrences. The main considerations are the choice of the vocabulary, the definition of words (treatment of compound words and acronyms), and the choice of the LM back-off strategy. There is, however, a significant amount of effort needed to process (or normalise) the texts before they can be used. One motivation for the normalisation is to reduce lexical variability so as to increase the coverage for a fixed size task vocabulary. The processing decisions are generally language specific (Davis, 2000).

Numerical expressions and dates are typically expanded to approximate the spoken form and to reduce the lexical variety (e.g. $150 in Shona becomes “*Zana nemakumi mashanu emadhora*”). Because of Shona's dependence on English for technical and quantitative nouns and verbs, it implies that we should factor in the English language model as well. Such dependence can be observed in statements like “year 1991”. In English this phrase is read as “The year nineteen ninety one” or “The year one thousand nine hundred and ninety one”. On the other hand in Shona, this phrase would be “Mugore ra nineteen ninety one” or “Mugore re churu ne makore mazana mapfumbamwe ne makore makumi mapfumbamwe ne rimwe”. The latter phrase is hardly used in contemporary spoken Shona. Therefore, in this research we used the first translation. To factor in the rule probabilities, we mainly consider the English language because of two reasons. Firstly, there are some researches that have been done that generalise the English language's probabilities of word sequences; hence we draw facts from these researches. Secondly, the Shona language borrows a considerable content from the English language. There are limitations associated with the approach we have taken. The main limitation is that the resulting system may miss some words during recognition because the language model being used is merged between Shona and English. The advantage however, is that, the resulting system can operate with the contemporary spoken Shona language.

Further semi-automatic processing is necessary to correct frequent errors inherent in the text (such as misspellings) or arising from processing with the distributed text processing tools. Some normalisation can be considered as “de-compounding” rules in that they modify the word boundaries and the total number of words. These concern the processing of ambiguous punctuation markers (such as hyphen and apostrophe), the processing of digit strings, and treatment of abbreviations and acronyms (ABCD can be transformed to A. B. C. D.). Shona is an agglutinative language (Kahari, 2001). De-compounding rules can be used to reduce the lexical variety. For example the phrase “I once walked past here”, in Shona becomes “Ndakambopafamba pano”. This phrase can be transformed into a word sequence “Ndaka mbo pa famba pano”. Depending upon the context, the recogniser hypotheses may need to be mapped into a more appropriate written form. Other normalisations (such as sentence initial capitalisation and case distinction) keep the total number of words unchanged, but reduce grapheme variability. In general a compromise is made between producing an output close to the standard written form of the language and the lexical coverage.

## Vocabulary Selection

Careful selection of the recognition vocabulary is important since on average, each out-of-vocabulary word causes more than one error, usually between 1.5 and 2 errors, (Gururaj et. al., 2003). We need to design the speech recogniser vocabulary with the goal of maximising lexical coverage for the expected input. A straightforward approach is to choose the N most frequent words in the training data which means the usefulness of the vocabulary is highly dependent upon the coverage of training data. To reduce this dependency, it is common practice to select a word list suited to the expected test conditions by minimising the system's out-of-vocabulary (OOV) rate on the LM development data. Therefore, judicious selection of the development data is important. The best lexical coverage may be obtained by selecting the vocabulary using only a subset of the training data (such as the most recent data) instead of using all available data. To reduce the error rate due to OOVs, the size of the lexicon can be increased. Using a very large lexicon has been shown to improve the performance, despite the potential of increased confusion with lexical entries (Gururaj et. al. 2003).

## N-gram Estimation

Using the Maximum Likelihood (ML) criterion, the n-gram probabilities are estimated from the frequencies of the word sequences of length n in the training corpus (texts or speech inscriptions). For example, the ML estimate of the trigram probability is given by:

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})}$$

where C(•) denotes the number of times the n-gram appears in the training data.

For large vocabulary sizes, many of the possible n-grams will not occur in even a very large training corpus. Due to the sparseness of the data, maximum likelihood estimates are clearly inadequate and need to be smoothed. Different approaches have been investigated to smooth the estimates of the probabilities of rare n-grams. The most common approach is to use a back-off mechanism which relies on a lower order n-gram (Hoffman et. al., 2006). If there is not enough data to obtain a robust estimate from the n-gram counts, a fraction of the probability mass is taken from the observed n-grams by discounting the ML estimates (Hoffman et. al., 1996). The probabilities of the rare n-grams are then estimated from the (n – 1)-gram probabilities in a recursive manner as shown here for a trigram model:

where B($w_{i-2}$, $w_{i-1}$) is a back-off coefficient needed to ensure that the probability sum for a given context is equal to one. Computing the bigram estimate P($w_i$ | $w_{i-1}$) follows the same principle. Backing-off offers an additional advantage in that the language model size can be arbitrarily reduced by increasing the cut-off frequencies below which the n-grams are not included in the model. This property can be used to reduce the amount of computational resources required during decoding. While 2-gram and 3-gram LMs are most widely used, small improvements can be obtained with the use of longer span LMs such as 4-grams and 5-grams (Schwarz, 2007). It is often the case that LM training corpus is comprised of different sources of texts of different sizes and in different formats. Model interpolation is an easy way to combine training material from different sources. A language model is trained for each source and the resulting models are interpolated. The interpolation weights can be directly estimated on some development data with the LM algorithm. An alternative approach is to simply merge the n-gram counts and train a single language model on these counts. However, other modelling techniques such as decision tree models, maximum entropy models, or linguistically motivated models (probabilistic context-free and link grammars), have been used with moderate success leading to small gains over the much simpler n-gram model (Schwarz, 2007)

$$\hat{P}(w_i|w_{i-2}, w_{i-1}) = \hat{P}(w_i|w_{i-1})\mathrm{B}(w_{i-2}, w_{i-1}),$$

## LM Adaptation

LVCSP systems use one or more language models, but these LMs are usually static, even though the choice of which model to use can be dynamic, dependent for example, on the dialogue state. Language Model adaptation is one of the interests for improving the model accuracy and for keeping the models up-to-date. In this research we use two language models, one for English and the other for Shona. The reason for this choice was explained in section 3.2. Various approaches have been taken to adapt the language model based on the observed text so far, including the use of a cache model, a trigger model or topic coherence model. The cache model is based on the idea that words appearing in the document will have an increased probability of appearing again in the same document. For small documents, the number of words appearing is limited, and as a consequence the benefit is very small. The trigger model attempts to overcome this by increasing the probabilities of words that often co-occur with the trigger word when the trigger word has been observed. In topic coherence modelling, selected keywords in the transcribed speech are used to retrieve articles on the similar topic with which sub-language models are constructed and used to re-score hypotheses. Despite the growing interest in adaptive language models, thus far, only minimal improvements have been obtained compared to the use of very large, static n-gram models (Sundberg, 2006).

## Pronunciation Modelling

The pronunciation dictionary is the link between the acoustic-level representation and the lexical items output by the speech recogniser. The accuracy of the acoustic models is partly dependent upon the consistency of the pronunciation dictionary. Associated with each lexical entry are one or more pronunciations, described using the chosen elementary units (usually phonemes or phones) this set of units is evidently language dependent. For example, some commonly used phones are 45 for English and 50 for Shona (Kahari, 2001). In generating pronunciation base-forms, most lexicons include standard full-form pronunciations and do not explicitly represent phonetic variants. This representation is chosen as most variants can be predicted by rules and their use is optional. More importantly, there often is a continuum between different phonetic realisation of a given phoneme and the decision as to which occurred in any given utterance is subjective. By using a phone representation, no hard decision is imposed, and it is left to the acoustic models to represent the observed variants in the training data. While pronunciation lexicons are usually (at least partially) created manually, several approaches to automatically learn and generate word pronunciations have been investigated. Such approaches, while promising have to date given only small performance improvements even when trained on manual transcriptions (Xuedong et. al., 2001).

Pronunciation variants can be observed for a variety of words. Alternative pronunciations are needed for homographs (words spelt the same, but pronounced differently) which reflect different parts of speech. Using a set of allophones models, the pronunciation probabilities are estimated by first aligning the reference word transcription with the audio signal (using a lexicon containing equally likely alternative pronunciations), letting the Viterbi algorithm choose the best pronunciation for each word. The probabilities are then estimated from the relative frequencies of each variant. Words of foreign origin, particularly, proper names may have different pronunciation depending upon the speaker's familiarity with the original language. It is also common for multisyllabic words to be pronounced with different numbers of syllables. If the acoustic model training is carried out without allowing for appropriate pronunciation variants, there will necessarily be a misalignment of one or more phones, making the phone models less accurate. Experience has shown that careful lexical design improves speech recognition system performance (Pentland et. al., 1995). In speech from fast speakers, or speakers with relaxed speaking styles it is common to observe poorly articulated (or skipped) unstressed syllables, particularly in long words with sequences of unstressed syllables. Although such long words are typically well recognised, often a nearby function word is deleted. To reduce these kinds of errors, alternate pronunciations in the lexicon can allow syllabic consonants in unstressed syllables. Compound words have also been used as a way to represent reduced forms for common word sequences. Fluent speech effects can alternatively be modelled using phonological rules. The principle behind the phonological rules is to modify the allowable phone sequences to take into account expected variations. These rules are optionally applied during training and recognition. Using phonological rules during training results in better acoustic models, since they will less “polluted” by wrong transcriptions. Their use during recognition reduces the number of mismatches (Pentland et. al. 1995). As speech recognition research has moved from read speech to found audio data, the phone set has been expanded to include non-speech events. These can correspond to noises produced by the speaker (e.g. breath noise, coughing, sneezing, laughter, etc.) or can correspond to external sources (music, motor, tapping, etc). In this research, we limited the scope to speech in a low noise environment.

## Acoustic Modelling

One of the main challenges of acoustic modelling is to handle the variability present in the speech signal. Variability can arise from the linguistic context, or can be associated with the non-linguistic context such as the speaker (e.g., the physical characteristics, speaking style, mood, etc.) and the acoustic environment (e.g., background noise, music) and the recording channel (e.g., direct microphone, telephone). We use Hidden Markov Models (HMMs) for acoustic modelling, which consists of modelling the probability density function of a sequence of acoustic feature vectors. Other approaches include segment based models and neural networks to estimate the acoustic observation likelihoods. With exception of the acoustic likelihood computation, the HMM framework is the most appropriate to combine linguistic and acoustic information in a single network representing all possible sentences (Baker, 1995).

## Acoustic Front-end

The first step of the acoustic feature analysis is digitisation, or conversion of the continuous speech signal into discrete samples. The most commonly used sampling rates are 16KHz and 10KHz for direct microphone input and 8KHz for telephone signals (Baker, 1995). The next step is feature extraction (also called front-end analysis), which has the goal of representing the audio signal in a more compact manner by trying to remove redundancy and reduce variability, while keeping the important linguistic information. An inherent assumption is that although the speech signal is continually changing, due to physical constraints on the rate at which the articulators can move, the signal can be considered quasi-stationary for short periods (on the order of 10 to 20ms). The most applicable set of features are cepstrum coefficients obtained with Mel Frequency Cepstral (MFC) analysis or the with a Perceptual Linear Prediction (PLP) analysis (Sundberg, 2006). Cepstral parameters are less correlated than direct spectral components, which simplifies estimation of the acoustic model parameters by reducing the need for modelling the dependency between features. In both cases a Mel scale short term power spectrum is estimated on a fixed window (typically in the range of 20 to 30ms). In order to avoid spurious high frequency components in the spectrum due to discontinuities caused by windowing the signal, it is common to use a tapered window such as a Hamming window (Sundberg, 2006). The window is then shifted, and the next feature vector computed. The most commonly used offset is 10ms. This acoustic parametrisation converts the speech signal into a sequence of feature vectors X, each vector representing a 10ms interval referred to as a frame or a feature vector:

$$X = (x_1, x_2, ..., x_T).$$

The Mel scale approximates the frequency resolution of the human auditory system, being linear in the low frequency range (below 1000Hz) and logarithmic above 1000Hz. The cepstral parameters are obtained by taking an inverse transform of the log of the filterbank parameters. In the case of the MFC coefficients, a cosine transform is applied to the log power spectrum, whereas a root Linear Predictive Coding (LPC) analysis is used to obtain the PLP cepstrum coefficients (Sundberg, 2006). Both sets of features can be used successfully, but PLP analysis has been found to be slightly more robust in the presence of background noise (Smits and Yegnanarayana, 1995). Cepstral mean removal (subtraction of the mean from all input frames, generally sentence based) is often used to reduce the dependency on the acoustic recording conditions. Computing the cepstral mean requires that all of the signal is available prior to processing, which is not the feasible in our case because processing needs to be synchronous with recording. In this case, a modified form of cepstral subtraction can be carried out where a running mean is computed from the N last frames (N is often on the order of 100), corresponding to 1s of speech. It is also common to normalise the feature variance, so that each resulting cepstral coefficient has a unity variance

(Smits and Yegnanarayana, 1995). In order to capture the dynamic nature of the speech signal, the feature vector is usually augmented with “delta” parameters. The delta parameters are computed by taking the first and second differences of the features in successive frames. As a result, a typically feature vector $x_t$ will include 12 cepstrum coefficients plus the normalised log-energy, along with the first and second order derivatives, i.e., a total of 39 components. Instead of using these fixed delta features, linear discriminant transforms are sometimes used to better optimise the feature vector for the acoustic models. Vocal Tract Length Normalisation (VTLN), a technique which performs a simple speaker normalisation at the front-end level, can also be used. The normalisation consists of performing a frequency warping to account for differences in vocal tract length, where the appropriate warping factor is chosen from a set of candidate values by maximising the test data likelihood based on a first decoding pass transcription and some acoustic models (Story et. al., 1996). VTLN must also be be applied during the training process to obtain models suited to decode the normalised test data. This normalisation has been shown to give significant error rate reduction in particular on telephone conversational speech.

## Modelling Allophones

Modelling allophones with Hidden Markov Models (HMMs) is appropriate because these models work reasonably well, and their parameters can be efficiently estimated using well established techniques. Allophone models offer a wide spectrum of contextual dependencies and back-off mechanisms to model rare contexts. The production of speech feature vectors is modelled in two steps. First a small Markov chain is used to generate a sequence of states and second, speech vectors are drawn using a probability density function (PDF) associated to each state. The Markov chain is described by the number of states and the transition probabilities between states. While different model topologies have been proposed, most make use of the left-to-right state sequences. The most commonly used configurations have 3 to 5 emitting states per allophone model, where the number of states imposes a minimal duration for the phone. Some configurations allow certain states to be skipped, thereby reducing the required minimal duration. The probability of an observation (i.e., a speech vector) is assumed to be dependent only on the current state. Given an N-state HMM with a parameter λ, the HMM stochastic process is described by the following joint probability density function of the observed signal

$$f(X,S|\lambda) = \pi_{s_0} \prod_{t=1}^{T} a_{s_{t-1}s_t} f(\mathbf{x}_t|s_t)$$

X = (x1, x2, x3, ..., xT) and the unobserved state sequence S = (s0, s1, s2, ..., sT),

where λi is the initial probability of state i, $a_{ij}$ is the transition probability from state i to state j and f(•|s) is the emitting PDF associated with each state s.

A powerful technique to keep the models trainable without sacrificing model resolution is to take advantage of the state similarity among different models of a given phone by tying the HMM state distributions. This basic idea is used in most current systems although there are slight differences in the implementation and in the naming of the resulting clustered states (senones, genones, PELs, tied-states). In practice both agglomerative clustering and divisive clustering have been found to yield model sets with comparable performance. Divisive decision tree clustering is particularly interesting when there are a very large number of states to cluster since it is at the same time both faster and more robust than a bottom up greedy algorithm, and therefore much easier to tune. In addition HMM state tying based on decision tree clustering has the advantage of providing the means to build models for unseen contexts, i.e., those contexts that do not occur in the training data. The set of questions typically concern the phone position, the distinctive features (and identities) of the phone and the neighbouring phones.

## Decoding

The LVCSP decoding problem is the design of an efficient search algorithm to deal with the huge search space obtained by combining the acoustic and language models. Strictly speaking, the aim of the decoder is to determine the most likely word sequence W*, given the language model, the pronunciation dictionary and the acoustic models. In practice, however, it is common to search for the most likely HMM state sequence. This maximum approximation, also referred to as the Viterbi search, leads to the simplified view of the decoding problem. This is an easier task, consisting of finding the best path through a trellis (the search space) where each node represents an HMM state at a given time. It has been shown that even though the Viterbi decoding gives only a crude approximation of the likelihood of the word sequence, the two word hypotheses are almost always very close (Schwarz, 2007). Some simple extensions of the Viterbi search are able to compensate for most of the decoding approximations in particular to avoid penalising words with many pronunciations. The first step of decoding is identifying the speech portions of the audio signal. We shall discuss this step in the next section.

## Speech/Non-Speech Detection

Detecting portions of the audio signal containing speech is commonly referred to as speech detection or end point detection. A variety of approaches to the endpoint detection have been proposed ranging from simple energy threshold based methods to methods requiring the extraction of more complex parameters such as pitch. A general view of the problem is one of data partitioning, which aims to divide a continuous audio stream into homogeneous acoustic segments. Partitioning consists of identifying speech and non-speech segments, and then clustering the speech segments, assigning metadata labels to each segment. The labels typically specify the signal bandwidth and gender, but can also specify the background characteristics and speaker identity. When transcribing inhomogeneous audio streams, partitioning the data prior to word recognition offers several advantages.

- In addition to the transcription of what was said, other interesting information can be extracted from the audio signal, such as the division into speaker turns and the speaker identities, and background acoustic conditions.
- By clustering segments from the same speaker, acoustic model adaptation can be carried out on a per cluster basis, as opposed to on a single segment basis, thus providing more adaptation data.
- Prior segmentation can avoid problems caused by linguistic discontinuity at speaker changes.
- By using acoustic models trained on particular acoustic conditions (such as wide-band or telephone band), overall performance can be significantly improved.
- Eliminating non-speech segments and dividing the data into shorter segments (which can still be several minutes long), substantially reduces the computation time and simplifies decoding.

Various approaches have been proposed to partition a continuous stream of audio data. Most of these approaches rely on a two step procedure, where the audio stream is first segmented in order to locate acoustic changes which are assumed to be associated with changes in the speaker, background or environmental condition, and channel condition (Schwarz, 2007). The segmentation procedures can be classified as being on phone decoding, distance-based segmentations or on hypotheses testing. The resulting segments are then clustered (usually using Gaussian Models), where each cluster is assumed to identify a speaker or more precisely, a speaker in a given acoustic condition. Alternative language-independent approach relies on an audio stream mixture model. Each component audio source, representing a speaker in a particular background and channel condition, is in turn modelled by a mixture of Gaussian models. The segment boundaries and labels are jointly identified via an iterative maximum likelihood segmentation/clustering procedure using Gaussian mixture models and agglomerative clustering.

## Decoding Strategies

Since it is often prohibitive to exhaustively search for the best path, techniques have been developed to reduce the computational load by limiting the search to a small part of the search space. Even for research purposes, where real-time recognition is not needed there is a limit on computing resources (memory and CPU time) above which the development process becomes too costly. The most commonly used approach for small and medium vocabulary sizes is the one-pass frame-synchronous Viterbi beam search which relies on a dynamic programming algorithm. This basic strategy has been extended to deal with large vocabularies by adding features such as dynamic decoding, multi-pass search, and N-best re-scoring. Dynamic decoding can be combined with efficient pruning techniques in order to obtain a single pass decoder that can provide the answer using all the available information (i.e., that is in the models) in a single forward decoding pass over of the speech signal. This kind of decoder, such as stack decoder is based on the A* algorithm or the one-pass frame synchronous dynamic network decoder, is very very attractive for real-time applications. Static decoders require much more memory than dynamic decoders when used with long span language models (3-gram or higher order), and as a consequence they are mostly used with smaller language models (usually 2-grams or constrained grammars). It has been recently shown that by proper optimisation of a finite-state automation corresponding to a recogniser HMM network, substantial reduction of the overall network size can be obtained, enabling static decoding with long span LM. However, the size of the optimised network remains proportional to the LM size. Multi-pass decoding can be used to progressively add knowledge sources in the decoding process, thus allowing the complexity of the individual decoding passes to be reduced and often resulting in a faster overall decoder. For example, a first decoding pass can use a 2-gram language model and simple acoustic models and later passes will make use of 3-gram and 4-gram language models with more complex acoustic models. This multiple pass paradigm requires a proper interface between passes in order to avoid losing information and engendering search errors. Information is usually transmitted via word lattices or word graphs. An HMM based speech recogniser can be seen as a transduction cascade which converts the observed feature vectors to a word string, where to some approximation, each transduction (phone model, word model or language model) can be represented as a finite state automation. Lattices are graphs where nodes correspond to particular frames and where edges representing word hypothesis have associated acoustic and language model scores.

It can sometimes be difficult to add certain knowledge sources into the decoding process especially when they do not fit the Markovian framework. This is the case when trying to use segmental information or to use grammatical information for long term agreement. Such information can be more easily integrated in a multi-pass system by re-scoring the recogniser hypotheses after applying the additional knowledge sources. Evidently, the first pass used to generate the initial word lattice must be accurate enough to not introduce lattice errors which are unrecoverable with further processing. In addition to multiple pass decoding, word lattices can be used to overcome the Viterbi approximation discussed above. As a matter of fact, true MAP decoding is a considerably easier task on a word lattice than on the original search space. Along the same lines, it has been proposed to use word lattices to perform a word based MAP decoding instead of word sequence MAP decoding, i.e., minimising the word error instead of the word sequence (or sentence) error rate (Davis, 2000).

## RESEARCH METHODOLOGY

In fundamental, the speech voice is a set of signals that concatenate voiced and unvoiced segments. The voiced segments can be considered as the result of a glottal pulse sequence filtered by the vocal tract. The pulse periodicity is never constant due to the fact that the voice organ is a complex mechanical system with many physical variables constantly evolving. In this sense, the periodicity might be stable up to a point where we can consider them as quasi-sinusoidal signals, but not of pure sinusoidal signals.

It is therefore not straightforward to model voice signals with just sinusoids, even when the phonation has the least possible aspirated air. In order to obtain a frequency domain resolution good enough as to distinguish the different frequency components (i.e. harmonics) and reliably estimate their parameters, we need a *window* covering at few periods of the signal. However, since the period is not constant neither the vocal tract immobile, we are then analysing together periods with different durations, and the magnitude spectrum is not a perfect train of pulses located at the fundamental period and its multiples. Indeed, covering several periods with the analysis window typically results in smearing and smoothing of the synthesized signal. This is probably the main difficulty of generating a model of the speech voice based on spectral harmonic models: the input signal contains non-stationary sinusoidal components and the accurate estimation of their parameters in real world signals becomes a challenge. Moreover, it is difficult to preserve the inner differences between consecutive pulses. Several techniques have been developed with aim of improving the accuracy in the estimation of each harmonic component, and to achieve the best possible trade-off between temporal and frequency resolution. Voiced utterances can be interpreted as:

- a set of time-varying quasi-sinusoidal signals looking at the frequency axis, or
- a sequence of voice pulses looking at the temporal axis.

This duality is illustrated in Figure 1. In the first case (a), we can consider voice utterances as a set of oscillators whose frequency relation is fixed to natural numbers (multiples of the fundamental frequency, inverse of the period), and whose amplitude is set by the target spectral envelope. Conversely, in the second case (b) and (c), we can consider voice utterances as a filtered train of time domain pulses at the target periodicity. A common problem of the first interpretation (a) is the one of shape variance: the loss of the intrinsic phase synchronisation between harmonics found in voice signals, also called phase-coherence.

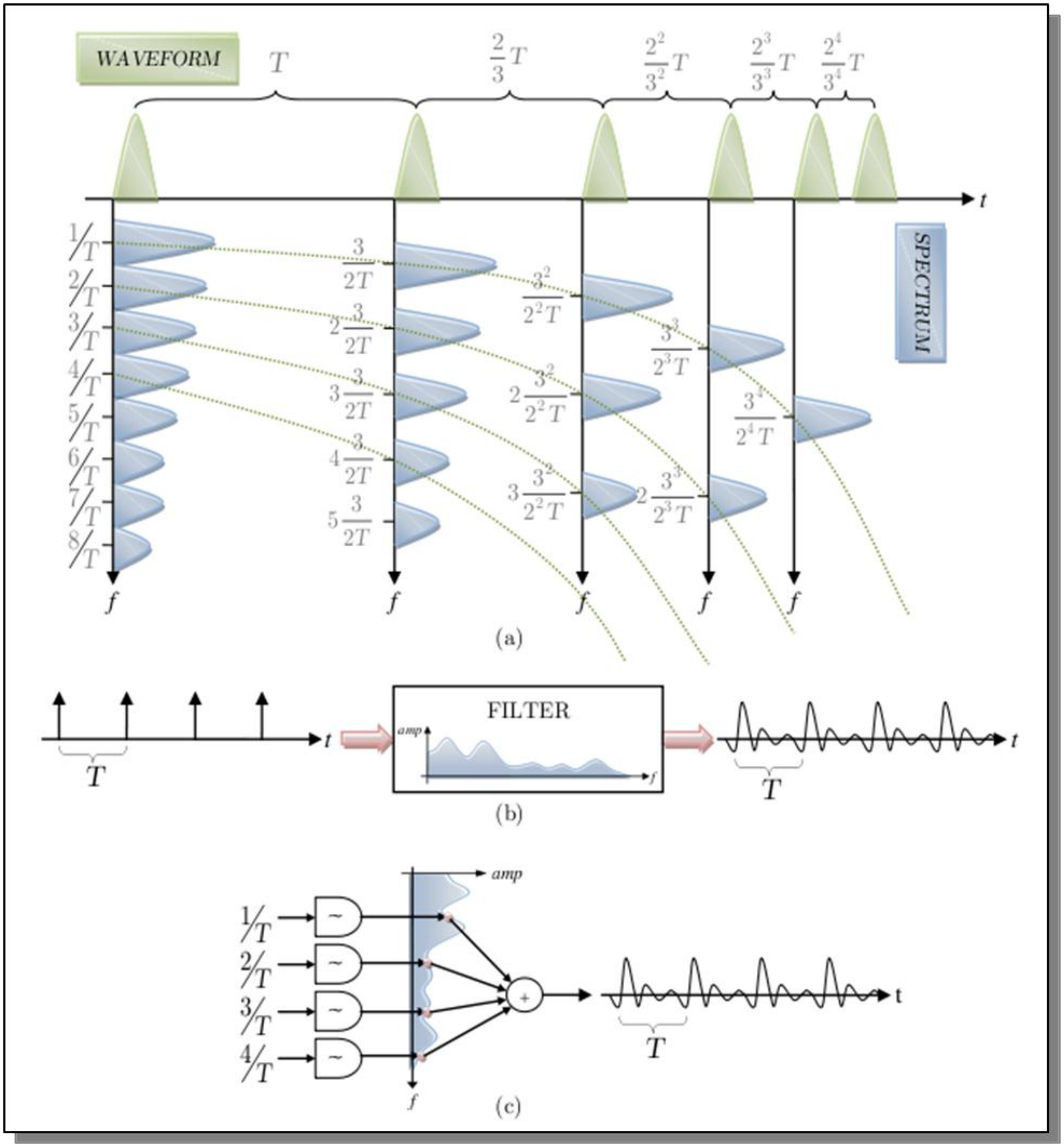


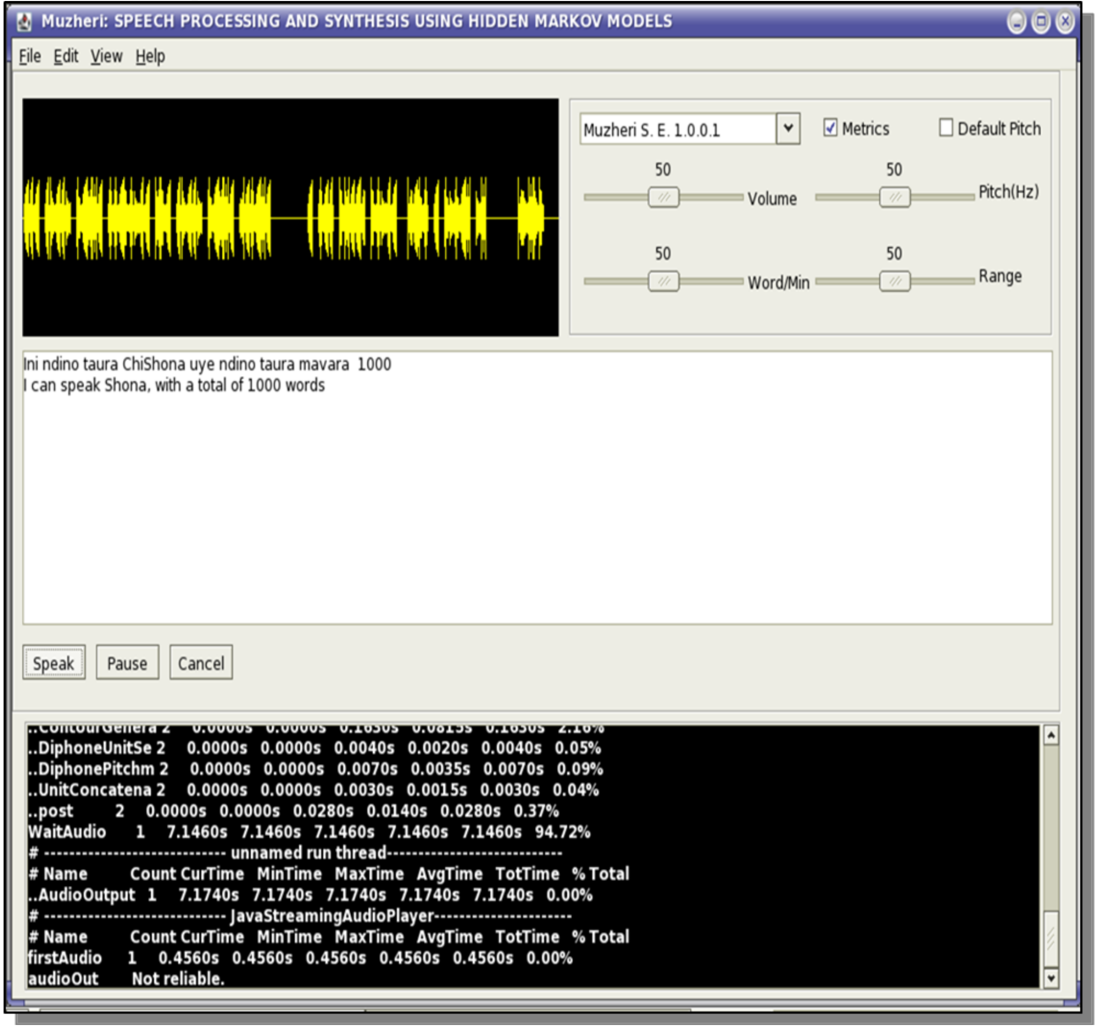
**Figure 1: Duality Nature of Voice Utterance Signals**

Harmonic signals can be seen as a set of time-varying quasi-sinusoidal signals (looking at the frequency axis), or as a sequence of voice pulses (looking at the time axis). In figure 1, (a) shows a train of time domain pulses (horizontal) with a period decreasing 2/3 each time, and its magnitude spectrum (vertical) at different time positions. It follows therefore that the inverse behaviour of time and frequency, producing a given temporal periodicity an inverse value of frequency periodicity (e.g. T seconds → 1/T Hertz). In the same way, while temporal pulses get closer, frequency components get more distant. The middle figure (b) shows the horizontal interpretation as a train of filtered pulses. In (c) we see the vertical interpretation as a group of oscillators whose amplitude follows the target spectral envelope. In this research we did not considered phase for simplification.

# IMPLEMENTATION AND TESTING

In this section we present the implementation and testing of the system. During the start-up process the systems attempts to access the input/output devices (e.g. microphone and speakers). It does not lock down these devices but attempts to gain reading and writing rights to and from these devices as may be necessary, hence these devices may be shared.

After a successful start up process, the system is ready to synthesise text to speech. This system has the capability of converting both the Shona and English words to speech. Figure 2 shows a screen shot of the system in the conversion process:



Ini ndino taura ChiShona uye ndino taura mavara  1000
I can speak Shona, with a total of 1000 words

```
..ContourGenera 2   0.0000s  0.0000s  0.1630s  0.0815s  0.1630s  2.16%
..DiphoneUnitSe 2   0.0000s  0.0000s  0.0040s  0.0020s  0.0040s  0.05%
..DiphonePitchm 2   0.0000s  0.0000s  0.0070s  0.0035s  0.0070s  0.09%
..UnitConcatena 2   0.0000s  0.0000s  0.0030s  0.0015s  0.0030s  0.04%
..post      2   0.0000s  0.0000s  0.0280s  0.0140s  0.0280s  0.37%
WaitAudio     1   7.1460s  7.1460s  7.1460s  7.1460s  7.1460s  94.72%
# --------------------------- unnamed run thread---------------------------
# Name       Count CurTime  MinTime  MaxTime  AvgTime  TotTime  % Total
..AudioOutput  1   7.1740s  7.1740s  7.1740s  7.1740s  7.1740s  0.00%
# --------------------------- JavaStreamingAudioPlayer--------------------
# Name       Count CurTime  MinTime  MaxTime  AvgTime  TotTime  % Total
firstAudio    1   0.4560s  0.4560s  0.4560s  0.4560s  0.4560s  0.00%
audioOut      Not reliable.
```

**Figure 2: Text-to-Speech Conversion**

In the Figure 2 screen shot, the system converts two statements to speech. The first statement is in Shona while the second statement is in English. The system uses the HMMs to estimate the mapping of the word in question. Some words may not be pronounced correctly but this can be improved through system training.

The system can convert spoken words to text. It can convert both Shona and English words. The system uses HMMs to estimate the most likely spoken word given the digitised voice sample. The word error rate (WER) may be reduced, hence increasing accuracy, through training the system.

The ability of the system to convert speech to text is based on digitisation of the speech signal and isolation of frequencies constituting the speech signal. The sonic space is therefore created when several frequencies in a speech signal can be captured and separated for analysis and reuse. 5.1 **Training**

Training the system to better recognise words and phrases is a process of adjusting parameters of the HMM's transitional probabilities. The transitional probability is the probability that the query word will be chosen from text corpus given the state of the phoneme. This depends on the emission probabilities for each of the words in each state. In this implementation, training can be done separately from the running of the system. This is achieved by adjusting the speech and text corpora.

During the progress of this research, different variants of statistical tests were implemented on both the text-to-speech and speech-to-text components of the system.

Although both text and speech input are converted with good quality on the average by the statistical approach, there were incidents of word and syntactic error. These errors are largely related to long range dependences on the n-gram language model used. To cope with these problems, we recommend morpho-syntactic analysis and grammar based language models for further research.

# DISCUSSION

Speech recognition is concerned with converting the speech waveform, an acoustic signal, into a sequence of words. We used the Shona language as the targeted language for processing and synthesis.While speech synthesis is concerned with converting a series of text words into a speech sound. Today's most practical approaches are based on a statistical modelling of the speech signal. In this chapter we present and address the questions of large vocabulary speech recognition, that is: language modelling, lexical representation, acoustic-phonetic modelling and decoding. For over a decade large vocabulary, continuous speech recognition has been one of the focal areas of research in speech recognition, serving as a test bed to evaluate models and algorithms (Davis, 2000). This chapter focuses on the HMMs as we develop methods to build a speaker-independent Large Vocabulary Continuous Speech Processing (LVCSP) system.

From a statistical point of view, speech is assumed to be generated by a language model which provides estimates of Pr(W) for all possible word strings W = ($w_1$, $w_2$, $w_3$, ....), and an acoustic model represented by a probability density function f(X|W) encoding the message W in the signal X. The goal of speech recognition is generally defined as finding the most likely word sequence given the observed acoustic signal, i.e., of maximising the probability of W given the speech signal X, or equivalently, maximising the product Pr(W)f(X|W). LVCSP systems use acoustic units corresponding to phones or phones-in-context, where each word is described by one or more phone transcriptions. Assuming that the speech signal X depends only on the underlying phone sequence H=($h_1$, $h_2$, $h_3$, ...), then f(X|W) can be rewritten as $\sum_H$Pr(H|W)f(X|H) where the summation is taken over the set pronunciations corresponding to the word sequence W. In practice this set is reasonably small as the average number of pronunciation variations per word is less than two. The underlying speech generation model is illustrated in Figure 4.1. The word sequence produced by the language model is successively transformed by two transducers, the pronunciation model and the acoustic model, to yield the speech signal. The formulation of the LVCSP problem leads to the following four main considerations:

- o **The language modelling problem:** Computing the priori probability Pr(W). It is usually estimated from relative n-gram frequencies in transcriptions of speech data as well as related text corpora.
- o **The pronunciation modelling problem:** The computation of Pr(H|W). This relies on a pronunciation dictionary which may include estimates of the word pronunciation probabilities.
- o **The Acoustic Modelling Problem:** Determining the structure of the probability density function f(X|H) and estimating its statistical parameters from speech samples. The most predominant approaches uses continuous density Hidden Markov Models (HMM) to represent context-dependent phones

- **The search problem**, i.e., determining the best word hypothesis for the speech data given the models. This is a big challenge for LVCSP systems due to the large vocabulary and language model size.

We used *phone* to refer to acoustic units without attempting to label them as phonemic (referring to the elementary and distinctive sounds in the language) or phonetic (the observed realisation of the elementary sounds). Contextual phone units (phones-in-context) implicitly model what can be considered *allophones*, i.e., contextual phonetic variants of the underlying phoneme.

## CONCLUSION

We employed Hidden Markov Models to process shona from voice data signals to text data and vice versa. A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be considered as the simplest dynamic Bayesian network. In a regular Markov Model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a Hidden Markov Model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Hidden Markov Models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bio-informatics.

We have contributed a new dimension to the design of Speech Processing and Synthesis Systems (SPSS). In our design the speech processing engine is separated from the top layer application. However, the control of intermediate probability values of phonemes may be controlled by the top layer application.

Shona Speech Synthesis and Recognition: The resulting system of this research can synthesise and recognise both English and Shona. We have successfully contributed a way to synthesise and recognise the Shona language which may possibly be the first. We built the the speech and text corpora that can be used for both Shona and English languages. We have contributed a way to improve speech recognition by introducing a method of capturing the speaker's expressivity. We do this by creating a sonic space through the separation of frequencies in a given speech signal. The use of the sonic space improves the recognition potential of the resulting system.

We recommend morpho-syntactic analysis and grammar based language models for further research. In linguistics, morphology is the identification, analysis and description of the structure of morphemes and other units of meaning in a language like words, affixes, and parts of speech and intonation/stress, implied context (words in a lexicon are the subject matter of lexicology). Taking this approach in speech recognition and synthesis would greatly reduce the word error rate hence increasing accuracy. When applying morpho-syntactic analysis and grammar based language models, the operationality criteria can be hand-crafted, or can be inferred from the treebank using either the entropy of its or-nodes. Hence the word error rate can be significantly reduced.

## REFERENCES

[1] Davies , K.H., Biddulph, R. and Balashek, S. (2002) Automatic Speech Recognition of Spoken Digits, J. Acoust. Soc. Am

[2] Davis RIA, Lovell BC (2000). "Comparing and evaluating HMM ensemble training algorithms using train and test and condition number criteria" (http://citeseer.ist.psu.edu/677948.html). Journal of Pattern Analysis and Applications

[3] Griffin E. G. "The History of Automatic Speech Recognition Evaluations at NIST". National Institute of Standards and Technology. May, 2009.NIST, 2009

[4] James K. Baker (1995). "The DRAGON System -- An Overview". IEEE Transactions on Acoustics Speech and Signal Processing 23: 24–29. doi:10.1109/TASSP.1975.1162650.

[5] Junqua, J, Haton, J. (2005). Robustness in Automatic Speech Recognition: Fundamentals and Applications. Kluwer Academic Publishers. ISBN 978-0792396468

[6] Kahari G., The Shona Language: Mambo Press, 2001

[7] Lawrence R. Rabiner (February 1989). "A tutorial on Hidden Markov Models and selected applications in speech recognition" (http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/ Reprints/ tutorial on hmm and applications. pdf). Proceedings of the IEEE 77 (2): 257–286.doi:10.1109/5. 18626. (http://www.cs.cornell.edu/courses/ cs481/2004fa/rabiner.pdf)

[8] Satish L, Gururaj BI (April 2003). " Use of hidden Markov models for partial discharge pattern classification (http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=212242)". IEEE Transactions on Dielectrics and Electrical Insulation

[9] Schwarz, D. "Corpus-based Concatenative Synthesis." IEEE Signal Processing Magazine, 2007: vol. 24, no. 1, January

[10] Smits, R., and B. Yegnanarayana. "Determination of Instants of Significant Excitation in Speech using Group Delay Function." IEEE Transactions on Speech and Audio Processing, 1995.

[11] Story, B.H., I.R. Titze, and E.A., Hoffman. "Vocal Tract Area Functions from Magnetic Resonance Imaging." Journal of Acoustics Society of America, 1996

[12] Story, B.H., I.R. Titze, and E.A., Hoffman. "Vocal Tract Area Functions from Magnetic Resonance Imaging." Journal of Acoustics Society of America, 1996

[13] Sundberg, J. "The KTH Synthesis of Singing." Advances in Cognitive Psychology, 2006

[14] Thad Starner, Alex Pentland. Visual Recognition of American Sign Language Using Hidden Markov (http:// citeseer.ist.psu.edu/starner95visual.html). Master's Thesis, MIT, Feb 1995, Program in Media Arts

[15] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon (2001). Spoken Language Processing. Prentice Hall. ISBN 0-13-022616-5.