Heiga Zen and Keiichi Tokuda

TechWare: HMM-Based Speech Synthesis Resources

Please send suggestions for Web resources of interest to our readers, proposals for columns, as well as general feedback, by e-mail to Dong Yu ("Best of the Web" associate editor) at dongyu@microsoft.com.

In this issue, "Best of the Web" focuses on hidden Markov model (HMM)-based speech synthesis, which has recently been demonstrated to be very effective in generating high-quality speech and started dominating speech synthesis research. The attractive point of this approach is that the synthesized speech can easily be modified by transforming HMM parameters with a small amount of speech data. Thus it is very useful for constructing speech synthesizers with various voice characteristics, speaking styles, and emotions.

This column first reviews the general speech synthesis technology and then describes HMM-based speech synthesis as well as some useful online resources. For the resources presented here, the attributes in square brackets describe the types of information. Unless otherwise noted, the resources are free.

SPEECH SYNTHESIS TECHNOLOGY

The development of computer-based speech synthesis technology has been ongoing for decades. In the early days, rule-based synthesis dominated the speech synthesis research. It generates synthetic speech by manipulating speech segments according to hand-crafted rules.

In 1990s, the speech synthesis technology progressed from the rule-based approach to the data-driven, corpus-based one. High-quality speech synthesizers can be built from sufficiently diverse single-speaker speech databases. We can see progress from fixed inventories, found in diphone synthesis, to the more general techniques of unit-selection synthesis, where appropriate subword units are selected from large databases. Unit-selection techniques evolved to be the dominant approach to speech synthesis. The following Wikipedia article is a good introduction to the history of speech synthesis technologies and related topics:

http://en.wikipedia.org/wiki/Speech_synthesis

Although certainly successful, there are always two limitations in unit-selection synthesis: i) when a given sentence requires phonetic and prosodic contexts not covered in a database, the quality of the synthesized speech can be severely degraded, and ii) as few modifications to the selected units are done, this limits the output speech to the same style as that in the original recordings. With the need for more control over speech variations, larger databases containing different styles are required. However, recording large databases with variations is costly.

HMM-BASED SPEECH SYNTHESIS

The idea of HMM-based speech synthesis first appeared in mid-1990s. It has been popular in speech synthesis research since the early 2000s. The basic procedure of HMM-based speech synthesis is as follows. We first extract parametric representations of speech including spectral (filter) and excitation (source) parameters from a speech database and then model them by using a set of subword HMMs. Usually, the maximum likelihood (ML) criterion is used to estimate the HMM parameters as

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} p(O \mid W, \boldsymbol{\lambda})$$

where $\boldsymbol{\lambda}$ is a set of HMM parameters, O is a set of training data, and W is a set of transcriptions corresponding to O. We then generate *most probable* speech parameters, $\hat{o}$, for a given sentence, w, from the set of estimated HMM parameters, $\hat{\boldsymbol{\lambda}}$, as

$$\hat{o} = \arg\max_{o} p(o \mid w, \hat{\boldsymbol{\lambda}}).$$

Finally, a speech waveform is reconstructed from the parametric representations of speech.

The HMM-based synthesis approach might be simply described as generating the *average* of similarly sounding speech segments. This contrasts with the need in unit-selection synthesis to retain unmodified speech units to synthesize speech waveforms. However, the use of parametric representation and statistical models offers other attractive points. The most attractive point on this approach is that by using only a small amount of speech data, we can easily build text-to-speech synthesis (TTS) systems with the target voice quality. Thus HMM-based synthesis is expected to be useful for constructing TTS systems with various flexibilities: voice characteristics, speaking styles, and emotions. There have been four major techniques to accomplish this, i.e., adaptation, interpolation, eigenvoice, and multiple regression. The following Web sites include speech samples that demonstrate the effects of these techniques.

SPEAKER/STYLE ADAPTATION

http://www.sp.nitech.ac.jp/index.php?HOME%2FDEMONSTRATION%2FSPEAKER%20ADAPTATION

http://www.kbys.ip.titech.ac.jp/demo/styleadapt/index.html

Digital Object Identifier 10.1109/MSP.2009.932563

SPEAKER/STYLE INTERPOLATION
http://www.sp.nitech.ac.jp/index.php?HOME%2FDEMONSTRATION%2FSPEAKER%20INTERPOLATION
http://homepages.inf.ed.ac.uk/jyamagis/Demo-html/SP-inter.html

EIGENVOICE
http://www.sp.nitech.ac.jp/index.php?HOME%2FDEMONSTRATION%2FEIGEN%20VOICE

MULTIPLE REGRESSION
http://www.kbys.ip.titech.ac.jp/demo/stylectrl/MRHSMM/index.html [demonstrations]

By using and combining these techniques, we can synthesize speech with various voice characteristics, speaking styles, and emotions without recording large speech databases.

Because of the space limitation, here we omit details of HMM-based synthesis. Please see the following tutorials for additional information:
http://www.sp.nitech.ac.jp/~tokuda/tokuda_iscslp2006.pdf
http://homepages.inf.ed.ac.uk/jyamagis/icassp/Lecture-HTS.pdf [tutorial slides]

SOFTWARE AND WEB RESOURCES

HMM TRAINING AND SYNTHESIS

HMM TOOLKIT
http://htk.eng.cam.ac.uk/

HMM-BASED SPEECH SYNTHESIS SYSTEM
http://hts.sp.nitech.ac.jp/

HTS_ENGINE
http://hts-engine.sourceforge.net/ [open-source, training and synthesis]

The HMM Toolkit (HTK) is a de-facto standard toolkit for training and manipulating HMMs in speech research. It consists of a set of libraries and tools in C and provides basis of HMM-based speech synthesis research.

The HMM-based speech synthesis system (HTS) is an extension of HTK that adds various functionalities for HMM-based speech synthesis. It is in C and released as a patch code to HTK. This site also releases recipes for building state-of-the-art speaker-dependent and speaker-adaptive synthesizers and some voices for the Festival Speech Synthesis System. The Publication page on this site lists a number of papers that are useful in understanding the HMM-based speech synthesis technology. There is also an open mailing list for the discussion of HTS (hts-users@sp.nitech.ac.jp). The Mailing List page archives the past messages and provides a search engine interface to explore them.

The hts_engine is a set of APIs of a run-time synthesis engine for HMM-based speech synthesis. It is also in C and provides various functionalities required to set up and drive the synthesis engine. Reference of the APIs and a stand-alone English TTS program for embedded devices are also released.

FRONT END

FESTIVAL SPEECH SYNTHESIS SYSTEM
http://www.cstr.ed.ac.uk/projects/festival/

MARY TEXT-TO-SPEECH
http://mary.dfki.de/

GALATEATALK
http://sourceforge.jp/projects/galateatalk/

FLITE
http://www.speech.cs.cmu.edu/flite/ [open-source, full TTS systems]

MECAB
http://sourceforge.net/projects/mecab/ [open-source, natural language processing]

Many TTS systems consist of two components: front end (text normalization, grapheme-to-phoneme conversion) and back end (waveform generation). The HMM-based speech synthesis performs the back-end part only. Therefore, it is essential to combine the HMM-based speech synthesis module with a front-end module in other software packages to build a complete TTS system.

The Festival Speech Synthesis System is a widely used general multilingual speech synthesis system in C++ and Scheme. The MARY TTS is a highly modular multilingual TTS system in Java. Both of them provide full TTS functionality with various APIs. The **hts_engine** has been incorporated as a module of Festival and MARY. GalateaTalk is an open-source Japanese TTS software. It also provides full TTS functionality and has its own HMM-based speech synthesis engine. Voices that are built by HTS can be used with this software without any other tools. Flite (festival-lite) is a small and fast run-time synthesis engine designed for embedded devices in C. On the hts_engine Web site, a small-footprint English TTS system consisting of hts_engine and Flite is released.

MeCab is an open-source fast and customizable morphological analyzer based on conditional random fields (CRFs) in C++. It is designed for generic purposes and can be applied to variety of TTS-related natural language processing tasks such as part-of-speech (POS) tagging, phrase chunking, and accent prediction.

SIGNAL PROCESSING

SPEECH SIGNAL PROCESSING TOOLKIT
http://sp-tk.sourceforge.net/

STRAIGHT
http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTtrial/
http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTadv/index_e.html

EDINBURGH SPEECH TOOLS
http://www.cstr.ed.ac.uk/projects/speech_tools/

SNACK SOUND TOOLKIT
http://www.speech.kth.se/snack/

ESPS SOFTWARE
http://www.speech.kth.se/software/#esps
http://www.ldc.upenn.edu/About/whatsnew.shtml [open source except STRAIGHT, signal processing]

The Speech Signal Processing Toolkit (SPTK) is a suite of speech signal processing tools in C and shell scripts. It includes (frequency-warped) linear predictive (LP) and cepstral analysis/synthesis, vector quantization, and other extended versions of them. We can use it to extract spectral and excitation parameters and

reconstruct a speech waveform from the spectral and excitation parameters generated from HMMs.

STRAIGHT is the high-quality analysis/synthesis method that can enhance the quality of HMM-based speech synthesis. The MATLAB p-code released on this site provides powerful fundamental frequency ($F0$) extraction, $F0$-adaptive spectral analysis, and aperiodicity measure extraction functions as well as waveform reconstruction function from extracted parameters. In the HTS recipes, the extracted parameters are converted to low-dimensional forms using SPTK and modeled by HMMs.

The Edinburgh Speech Tools is a collection of APIs and programs for speech processing including waveform manipulation, feature extraction, and conversion. It also includes a pitch tracker, a labeling system, a classification and regression tree (CART), and support for linguistic type objects. This tool is used in the Festival Speech Synthesis System.

Robust $F0$ extraction is essential in building a high-quality HMM-based speech synthesizer. The Snack Sound Toolkit is a Tcl/Tk or Python-based speech processing engine that provides two types of $F0$ extraction functionalities based on the average magnitude difference function (AMDF) and the robust algorithm for pitch tracking (RAPT, also known as get_f0). ESPS software is a suite of signal processing programs that can be used for the analysis, manipulation, and labeling of speech. It also includes the stand-alone version of get_f0.

DATABASE

http://festvox.org/cmu_arctic/
[speech synthesis corpus]

The CMU ARCTIC databases are phonetically balanced, U.S. English, single-speaker databases designed for speech synthesis research. The HTS recipes for building state-of-the-art, speaker-dependent and speaker-adaptive HTS voices use these databases.

TRANSCRIPTION EDITOR

http://www.speech.kth.se/wavesurfer/
[open-source, segmentation editor]

The use of manually corrected segmentation labels sometimes improves the quality of synthesized speech. Wavesurfer is an open-source GUI sound manipulation tool based on Snack. It can read and write HTK-style transcription files and provides an intuitive interface to manually edit segmentation labels.

SPEECH SYNTHESIS EVALUATION

http://festvox.org/blizzard
http://www.synsig.org/index.php/Blizzard_Challenge/
[workshop description and papers]

The Blizzard Challenge is an annual large-scale, open-speech synthesis evaluation where common speech databases are provided to participants to build synthetic voices. This page archives the papers presented in the Blizzard Challenge workshops, which is annually held as a satellite workshop of Interspeech. The HMM-based speech synthesis systems have demonstrated its potential to synthesize high-quality speech since the first challenge.

PROJECT

http://www.emime.org/
[EU-funded project]

The EMIME is an EU-funded project focusing on developing a mobile device that can perform speech-to-speech translation with user's voice characteristics. The HMM-based speech synthesis is used as a core technology in this project.

ONLINE TTS DEMOS

http://www.festvox.org/voicedemos.html
http://www.cstr.ed.ac.uk/projects/festival/morevoices.html
http://www.sp.nitech.ac.jp/~demo/gtalk/demo.php
http://homepages.inf.ed.ac.uk/jyamagis/Demo-html/demo.html

The festvox voice-demo page provides online demos of basic HTS voices. The Edinburgh University CSTR online demo page also provides TTS demos of the state-of-the-art HTS voices submitted to the past Blizzard Challenge events. An online Japanese HTS-based TTS demo can be found on the GalateaTalk demo page. Another Edinburgh University demo page provides various speech samples and online demos for the state-of-the-art applications and technologies of HMM-based speech synthesis.

AUTHORS

Heiga Zen (heiga.zen@crl.toshiba.co.uk) is a research engineer with the Speech Technology Group at the Toshiba Research Europe Ltd. Cambridge Research Laboratory, U.K.

Keiichi Tokuda (http://www.sp.nitech.ac.jp/~tokuda) is a professor with the Department of Computer Science at the Nagoya Institute of Technology, Japan.

[SP]