

Enrich Web Applications with Voice Internet Persona Text-to-Speech for Anyone, Anywhere

Min Chu, Yusheng Li, Xin Zou, and Frank Soong

Micorosoft Research Asia, Beijing, P.R.C., 100080
{minchu,yushli,xinz,frankkps}@microsoft.com

Abstract. To embrace the coming age of rich Internet applications and to enrich applications with voice, we propose a Voice Internet Persona (VIP) service. Unlike current text-to-speech (TTS) applications, in which users need to painstakingly install TTS engines in their own machines and do all customizations by themselves, our VIP service consists of a simple, easy-to-use platform that enables users to voice-empower their content, such as podcasts or voice greeting cards. We offer three user interfaces for users to create and tune new VIPs with built-in tools, share their VIPs via this new platform, and generate expressive speech content with selected VIPs. The goal of this work is to popularize TTS features to additional scenarios such as entertainment and gaming with the easy-to-access VIP platform.

Keywords: Voice Internet Persona, Text-to-Speech, Rich Internet Application.

1 Introduction

The field of text-to-speech (TTS) conversion has seen a great increase in both research community and commercial applications over the past decade. Recent progress in unit-selection speech synthesis [1-3] and Hidden Markov Model (HMM) speech synthesis [4-6] has led to considerably more natural-sounding synthetic speech that is suitable for many applications. However, only a small part of these applications have had TTS features. One of the key barriers for popularizing TTS in various applications is the technical difficulty in installing, maintaining and customizing a TTS engine. In this paper, we propose a TTS service platform, the *Voice Internet Persona* (VIP), which we hope will provide an easy-to-use platform for users to voice-empower their content or applications at any time and anywhere.

Currently, when a user wants to integrate TTS into an application, he has to search the engine providers, pick one from the available choices, buy a copy of the software, and install it on his machines. He or his team has to understand the software. The installing, maintaining and customizing of a TTS engine can be a tedious process. Once a user has made a choice of a TTS engine, he has limited flexibility in choosing voices. It is not easy to demand a new voice unless one wishes to pay for additional development costs. It is virtually impossible for an individual user to have multiple TTS engines with dozens or hundreds voices for use in applications. With the VIP platform, users will not be bothered by technical issues. All their operations would be

encompassed in the VIP platform, including selecting, employing, creating and managing the VIPs. Users could access the service when they require TTS features. They could browse or search the VIP pool to find the voice they like and use it in their applications, or easily change it to another VIP or use multiple VIPs in the same application. Users could even create their own private voices through a simple interface and built-in tools.

The target users of the VIP service include Web-based service providers such as voice greeting card companies, as well as numerous individual users who regularly or occasionally create voice content such as Podcasts or photo annotations.

This paper is organized as follows. In Section 2, the design philosophy is introduced. The architecture of the VIP platform is described in Section 3. In Section 4, the TTS technologies and voice-morphing technologies that would be used are introduced. A final discussion is in Section 5.

2 The Design Philosophy

In the VIP platform, multiple TTS engines are installed. Most of them have multiple built-in voices and support some voice-morphing algorithms. These resources are maintained and managed by the service provider. Users will not be involved in technical details such as choosing, installing, and maintaining TTS engines and would not have to worry about how many TTS engines were running and what morphing algorithms would be supported. All user-related operations would be organized around the core object — the VIP.

VIP is an object with many properties, including a greeting sentence, its gender, the age range it represents, the TTS engine it uses, the language it speaks, the base voice it is derived from, the morphing targets it supports, the morphing target that is applied, its parent VIP, its owner and popularity, etc. Each VIP has a unique name, through which users can access it in their applications. Some VIP properties are exposed to users in a *VIP name card* to help identify a particular VIP. New VIPs are easily derived from existing ones by inheriting main properties and overwriting some of them. Within the platform, there is a VIP pool that includes base VIPs to represent all base voices supported by all TTS engines and derived VIPs that are created by applying a morphing target on a base VIP.

The underlying voice-morphing algorithms are rather complicated because different TTS engines support different algorithms and there are many free parameters in each algorithm. Only a small portion of the possible combinations of all free parameters will generate meaningful morphing effects. It's too time-consuming to understand and master these parameters for most users. Instead, a set of morphing targets that is easily understood to users are designed. Each target is attached with several pre-tuned parameter sets, representing the morphing degree or directions. All technical details are hidden from users. What a user would do is pick up a morphing target and select a set of parameters. For example, users can increase or decrease the pitch level and the speech rate, can convert a female voice to a male voice or vice versa, convert a normal voice to a robot-like voice, add a venue effect such as *in the valley* or *under the sea*, or make a Mandarin Chinese voice render Ji'nan or Xi'an dialect. Users will hear a synthetic example immediately after each change in

morphing targets or parameters. Currently, four types of morphing targets, as listed in Table 1, are supported in the VIP platform. The technical details on morphing algorithms and parameters are introduced in Section 4.

Table 1. The morphing targets supported in the current VIP platform

Speaking style	Speaker	Accent from local dialect	Venue of speaking
Pitch level	Man-like	Ji'nan accent	Broadcast
Speech rate	Girl-like	Luoyang accent	Concert hall
Sound scared	Child-like	Xi'an accent	In valley
	Hoarse or Reedy	Southern accent	Under sea
	Bass-like		
	Robot-like		
	Foreigner-like		

The goal of the VIP service is to make TTS easily understood and accessible for anyone, anywhere so that more and more users would like to use Web applications with speech content. With this design philosophy, a VIP-centric architecture is designed to allow users to access, customize, and exchange VIPs.

3 Architecture of the VIP Platform

The architecture of the VIP platform is shown in Fig. 1. Users interact with the platform through three interfaces designed for employing, creating and managing VIPs. Only the VIP pool and the morphing target pool are exposed to users. Other resources like TTS engines and their voices are invisible to users and can only be accessed indirectly via VIPs. The architecture allows adding new voices, new languages, and new TTS engines. The three user interfaces are described in Sub-section 3.1 to 3.3 below and the underlying technologies in TTS and voice-morphing are introduced in Section 4.

3.1 VIP Employment Interface

The VIP employment interface is simple. Users insert a VIP name tag before the text they want spoken and the tag takes effect until the end of the text unless another tag is encountered. A sample script for creating speech with VIPs is shown in Table 2. After the tagged text is sent to the VIP platform, it is converted to speech with the appointed VIPs and the waveform is delivered back to the users. This is provided along with additional information such as the phonetic transcription of the speech and the phone boundaries aligned to the speech waveforms if they are required. Such information can be used to drive lip-syncing of a talking head or to visualize the speech and script in speech learning applications.

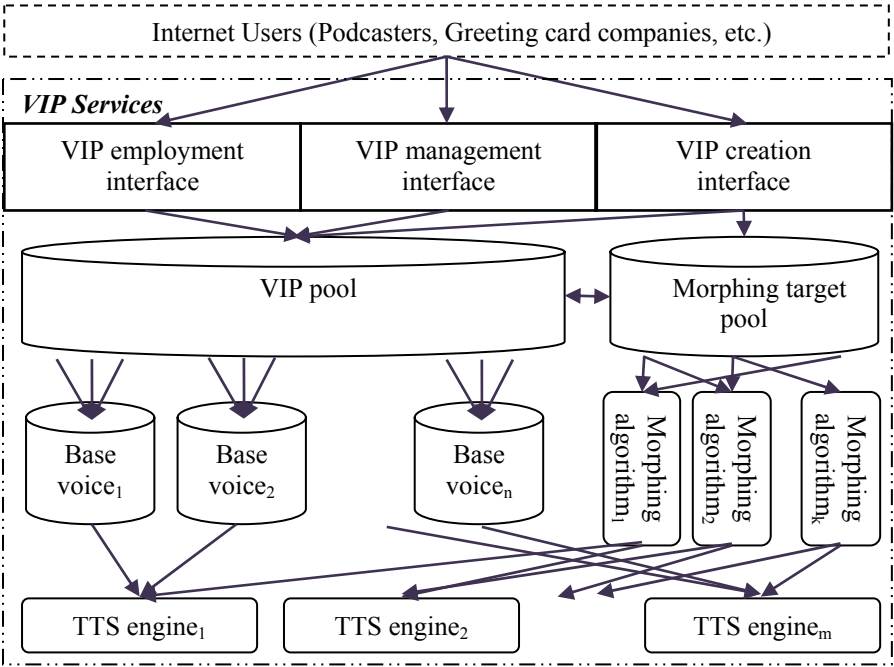


Fig. 1. Architecture of the VIP platform

3.2 VIP Creation Interface

Fig. 2 shows the VIP creation interface. The right window is the VIP view, which consists of a public VIP list and a private list. Users can browse or search the two lists, select a seed VIP and make a clone of one under a new name. The top window shows the name card of the focused VIP. Some properties in the view, such as gender and age range, can be directly modified by the creator. Others have to be overwritten through built-in functions. For example, when the user changes a morphing target, the corresponding field in the name card is adjusted accordingly. The large central window is the morphing view, showing all morphing targets and pre-tuned parameter sets. Users can choose one parameter set in one target as well as clear the morphing setting. After a user finishes the configuration of a new VIP, its name card is sent to the server for storage and the new VIP is shown in his private view.

3.3 VIP Management Interface

After a user creates a new VIP, the new VIP is only accessible to the creator unless the creator decides to share it with others. Through the VIP management interface, users can edit, group, delete, and share their private VIPs. User can also search VIPs by their properties, such as all female VIPs, VIPs for teenage or old men, etc.

Table 2. An example of the script for synthesis

- <persona mom> Hi, kids, let's annotate the pictures taken in our China trip and share them with grandpa through the Internet.
- <persona dad> OK. Lucy and David, we are connected to the VIP site now.
- <persona Lucy> This picture was taken at the Great Wall. Isn't it beautiful?
- <persona David> See, I am on top of a signal fire tower.
- <persona Lucy> This was with our Chinese tour guide, Lanlan. She knows all historic sites in Beijing very well.
- <persona Lanlan> This is the Summer Palace, the largest imperial park in Beijing. And here is the Center Court Area, where Dowager and Emperor used to met officials and conduct their state affairs.

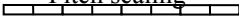
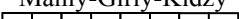
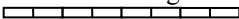
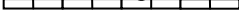
VIP view	VIP name card	
Private VIPs	Name: Dad; Gender: male; Age range: 30-50; Engine: Mulan; Voice: Tom; Language: English; Morphing applied: pitch scale; Parent VIP: Tom; Greeting words: Hello, welcome to use the VIP service	
Dad Mom Lucy Cat Robot	Morphing targets	
Public VIPs	Speaking style	Speaker
	Pitch scaling 	Manly-Girly-Kidzy 
	Rate scaling  Scared speech 	<input type="checkbox"/> Hoarse <input type="checkbox"/> Reedy <input checked="" type="checkbox"/> Bass <input type="checkbox"/> Robot <input type="checkbox"/> Foreigner
Anna Sam Tom Harry Lisa Lili Tongtong Jiajia	Chinese dialect <input type="checkbox"/> Ji'nan <input checked="" type="checkbox"/> Xi'an <input type="checkbox"/> Quoyang	Speaking venue <input type="checkbox"/> Broadcasted <input checked="" type="checkbox"/> Concert hall <input type="checkbox"/> In valley <input type="checkbox"/> Under sea

Fig. 2. The interface for creating new VIPs

4 Underlying Component Technologies

4.1 TTS Technologies

There are two TTS engines installed in the current deployment of VIP platform. One is Microsoft Mulan [7], a unit selection based system in which a sequence of waveform segments are selected from a large speech database by optimizing a cost function. These segments are then concatenated one-by-one to form a new utterance.

The other is an HMM-based system [8]. In this system, context dependent phone HMMs have been pre-trained from a speech corpus. In the run-time system, trajectories of spectral parameters and prosodic features are first generated with constraints from statistical models [5] and are then converted to a speech waveform.

4.2 Unit-Selection Based TTS

In a unit-selection based TTS system, naturalness of synthetic speech, to a great extent, depends on the goodness of the cost function as well as the quality of the unit inventory.

Cost Function

Normally, the cost function contains two components, the target cost, which estimates the difference between a database unit and a target unit, and the concatenation cost, which measures the mismatch across the joint boundary of consecutive units. The total cost of a sequence of speech units is the sum of the target costs and the concatenation costs.

In early work [2,9], acoustic measures, such as Mel Frequency Cepstrum Coefficients (MFCC), f_0 , power and duration, were used to measure the distance between two units of the same phone type. All units of the same phone are clustered by their acoustic similarity. The target cost for using a database unit in the given context is then defined as the distance of the unit to its cluster center, i.e. the cluster center is believed to represent the target values of acoustic features in the context. With such a definition for target cost, there is a connotative assumption, i.e. for any given text, there always exists a best acoustic realization in speech. However, this is not true in human speech. In [10], it was reported that even under highly restricted condition, i.e., when the same speaker reads the same set of sentences under the same instruction, rather large variations are still observed in phrasing sentences as well as in forming f_0 contours. Therefore, in Mulan, no f_0 and duration targets are predicted for a given text. Instead, contextual features (such as word position within a phrase, syllable position within a word, Part-of-Speech (POS) of a word, etc.) that have been used to predict f_0 and duration targets in conventional studies are used in calculating the target cost directly. The connotative assumption for this cost function is that speech units spoken in similar context are prosodically equivalent to one another in unit selection if we do have a suitable description of the context.

Since, in Mulan, speech units are always joint at phone boundaries, which are the rapid change areas of spectral features, the distances between spectral features at the two sides of the joint boundary is not an optimal measure for the goodness of concatenation. A rather simple concatenation cost is defined in [10]: the continuity for splicing two segments is quantized into four levels: 1) continuous — if two tokens are continuous segments in the unit inventory, the target cost is set to 0; 2) semi-continuous — though two tokens are not continuous in the unit inventory, the discontinuity at their boundary is often not perceptible, like splicing of two voiceless segments (such as /s/+t/), a small cost is assigned.; 3) weakly discontinuous — discontinuity across the concatenation boundary is often perceptible, yet not very strong, like the splicing between a voiced segment and an unvoiced segment (such as /s/+ a:/) or vice versa, a moderate cost is used; 4) strongly discontinuous — the

discontinuity across the splicing boundary is perceptible and annoying, like the splicing between voiced segments, a large cost is assigned. Type 1 and 2 are preferred in concatenation and the 4th type should be avoided as much as possible.

Unit Inventory

The goal of unit selection is to find a sequence of speech units that minimize the overall cost. High-quality speech will be generated only when the cost of the selected unit sequence is low enough [11]. In other words, only when the unit inventory is large enough so that we always can find a good enough unit sequence for a given text, we will get natural sounding speech. Therefore, creating a high-quality unit inventory is crucial for unit-selection based TTS systems.

The whole process of the collection and annotation of a speech corpus is rather complicated and contains plenty of minutiae that should be handled carefully. In fact, in many stages, human interference such as manually checking or labeling is necessary. Creating a high-quality TTS voice is not an easy task even for a professional team. That is why most state-of-the-art unit selection systems can provide only a few voices. In [12], a uniform paradigm for creating multi-lingual TTS voice databases with focuses on technologies that reduce the complexity and manual work load of the task has been proposed. With such a platform, adding new voices to Mulan becomes relatively easier. Many voices have been created from carefully designed and collected speech corpus (>10 hour of speech) as well as from some available audio resources such as audio books in the public domain. Besides, several personalized voices are built from small, office recording speech corpus, each consisting of about 300 carefully designed sentences read by our colleagues. The large foot-print voices sound rather natural in most situations, while the small ones sound acceptable only in specific domains.

The advantage of unit selection based approach is that all voices can reproduce the main characteristics of the original speakers, in both timber and speaking style. The disadvantages of such systems are that sentences containing unseen context will have discontinuity problem sometime and these systems have less flexibility in changing speakers, speaking styles or emotions. The discontinuity problem becomes more severe when the unit inventory is small.

4.3 HMM Based TTS

To achieve more flexibility in TTS systems, the HMM-based approach has been proposed [1-3]. In such a system, speech waveforms are represented by a source-filter model. Both excitation parameters and spectral parameters are modeled by context dependent HMMs. The training process is similar to that in speech recognition. The main difference lies in the description of context. In speech recognition, normally only the phones immediately before and after the current phone are considered. However, in speech synthesis, all context features that have been used in unit selection systems can be used. Besides, a set of state duration models are trained to capture the temporal structure of speech. To handle the data scarcity problem, a decision tree based clustering method is applied to tie context dependent HMMs. During synthesis, a given text is first converted to a sequence of context-dependent units in the same way as it is done in a unit-selection system. Then, a sentence HMM

is constructed by concatenating context-dependent unit models. Then, a sequence of speech parameters, including both spectral parameters and prosodic parameters, are generated by maximizing the output probability for the sentence HMM. Finally, these parameters are converted to a speech waveform through a source-filter synthesis model. In [3], mel-cepstral coefficients are used to represent speech spectrum. In our system [8], Line Spectrum Pair (LSP) coefficients are used.

The requirement for designing, collecting and labeling of speech corpus for training a HMM-based voice is almost the same as that for a unit-selection voice, except that the HMM voice can be trained from a relative small corpus and still maintains reasonably good quality. Therefore, all speech corpus used by the unit-selection system are used to train HMM voices.

Speech generated with the HMM system is normally stable and smooth. The parametric representation of speech gives us good flexibility to modify the speech. However, like all vocoded speech, speech generated from the HMM system often sounds buzzy. It is not easy to draw a simple conclusion on which approach is better, unit selection or HMM. In certain circumstance, one may outperform the other. Therefore, we installed both engines in the platform and delay the decision-making process to a time when users know better what they want do.

4.4 Voice-Morphing Algorithms

Three voice-morphing algorithms, sinusoidal-model based morphing, source-filter model based morphing and phonetic transition, are supported in this platform. Two of them seek to enable pitch, time and spectrum modifications and are used by the unit-selection based systems and HMM-based systems. The third one is designed for synthesis dialect accents with the standard voice in the unit selection based system.

4.5 Sinusoidal-Model Based Morphing

To achieve flexible pitch and spectrum modifications in unit-selection based TTS system, the first morphing algorithm is operated on the speech waveform generated by the TTS system. Internally, the speech waveforms are still converted into parameters through a Discrete Fourier Transforms. To avoid the difficulties in voice/unvoice detection and pitch tracking, a uniformed sinusoidal representation of speech, shown as in Eq. (1), is adopted.

$$S_i(n) = \sum_{l=1}^{L_i} A_l \cdot \cos[\omega_l n + \theta_l] \quad (1)$$

where A_l , ω_l and θ_l are the amplitudes, frequencies and phases of the sinusoidal components of speech signal $S_i(n)$, L_i is the number of components considered. These parameters are obtained as described in [13] and can be modified separately.

For pitch scaling, the central frequencies of all components are scaled up or down by the same factor simultaneously. Amplitudes of new components are sampled from the spectral envelop formed by interpolating A_l . All phrases are kept as before. For formant position adjustment, the spectral envelop forms by interpolating between

A_i is stretched or compressed toward the high-frequency end or the low-frequency end by a uniform factor. With this method, we can increase or decrease the formant frequencies together, yet we are not able to adjust the individual formant location. In the morphing algorithm, the phase of sinusoidal components can be set to random values to achieve whisper or hoarse speech. The amplitudes of even or odd components can be attenuated to achieve some special effects.

Proper combination of the modifications of different parameters will generate the desired style, speaker morphing targets listed in Table 1. For example, if we scale up the pitch by a factor 1.2-1.5 and stretch the spectral envelop by a factor 1.05-1.2, we are able to make a male voice sound like a female. If we scale down the pitch and set the random phase for all components, we will get a hoarse voice.

4.6 Source-Filter Model Based Morphing

Since in the HMM-based system, speech has been decomposed to excitation and spectral parameters. Pitch scaling and formant adjustment is easy to achieve by adjusting the frequency of excitation or spectral parameters directly. The random phase and even/odd component attenuation are not supported in this algorithm. Most morphing targets in style morphing and speaker morphing can be achieved with this algorithm.

4.7 Phonetic Transition

The key idea of phonetic transition is to synthesize closely related dialects with the standard voice by mapping the phonetic transcription in the standard language to that in the target dialect. This approach is valid only when the target dialect shares similar phonetic system with the standard language.

A rule-based mapping algorithm has been built to synthesize Ji'nan, Xi'an and Luoyang dialects in China with a Mandarin Chinese voice. It contains two parts, one for phone mapping, and the other for tone mapping. In the on-line system, the phonetic transition module is added after the text and prosody analysis. After the unit string in Mandarin is converted to a unit string representing the target dialect, the same unit selection is used to generate speech with the Mandarin unit inventory.

5 Discussions

The conventional TTS applications include call center, email reader, and voice reminder, etc. The goal of such applications is to convey messages. Therefore, in most state-of-the-art TTS systems, broadcast style voices are provided. With the coming age of rich internet applications, we would like to popularize TTS features to more scenarios such as entertainment, casual recording and gaming with our easy-to-access VIP platform. In these scenarios, users often have diverse requirements for voices and speech styles, which are hard to fulfill in the traditional way of using TTS software. With the VIP platform, we can incrementally add new TTS engines, new base voices and new morphing algorithms without affecting users. Such a system is able to provide users enough diversity in speakers, speaking styles and emotions.

In the current stage, new VIPs are created by applying voice-morphing algorithms on provided bases voices. In the next step, we will extend the support to build new voices from user-provided speech waveforms. We also look into opportunities to deliver voice in other applications via our programming interface.

References

1. Wang, W.J., Campbell, W.N., Iwahashi, N., Sagisaka, Y.: Tree-Based Unit Selection for English Speech Synthesis. In: Proc. of ICASSP-1993, Minneapolis, vol.2, pp. 191–194 (1993)
2. Hunt, A.J., Black, A.W.: Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database. In: Proc. of ICASSP- 1996, Atlanta, vol. 1, pp. 373–376 (1996)
3. Chu, M., Peng, H., Yang, H.Y., Chang, E.: Selecting Non-Uniform Units from a Very Large Corpus for Concatenative Speech Synthesizer. In: Proc. of ICASSP-2001, Salt Lake City, vol. 2, pp. 785–788 (2001)
4. Yoshimura, T., Tokuda, K., Masuku, T., Kobayashi, T., Kitamura, T.: Simultaneous Modeling Spectrum, Pitch and Duration in HMM-based Speech Synthesis. In: Proc. of European Conference on Speech Communication and Technology, Budapest, vol. 5, pp. 2347–2350
5. Tokuda, K., Kobayashi, T., Masuko, T., Kobayashi, T., Kitamura, T.: Speech Parameter Generation Algorithms for HMM-based Speech Synthesis. In: Proc. of ICASSP-2000, Istanbul, vol. 3, pp. 1315–1318 (2000)
6. Tokuda, K., Zen, H., Black, A.W.: An HMM-based Speech Synthesis System Applied to English. In: Proc. of 2002 IEEE Speech Synthesis Workshop, Santa Monica, pp. 11–13 (2002)
7. Chu, M., Peng, H., Zhao, Y., Niu, Z., Chang, E.: Microsoft Mulan — a bilingual TTS systems. In: Proc. of ICASSP-2003, Hong Kong, vol. 1, pp. 264–267 (2003)
8. Qian, Y., Soong, F., Chen, Y.N., Chu, M.: An HMM-Based Mandarin Chinese Text-to-Speech System. In: Huo, Q., Ma, B., Chng, E.-S., Li, H. (eds.) ISCSLP 2006. LNCS (LNAI), vol. 4274, pp. 223–232. Springer, Heidelberg (2006)
9. Black, A.W., Taylor, P.: Automatic Clustering Similar Units for Unit Selection in Speech Synthesis. In: Proc. of Eurospeech-1997, Rhodes, vol. 2, pp. 601–604 (1997)
10. Chu, M., Zhao, Y., Chang, E.: Modeling Stylized Invariance and Local Variability of Prosody in Text-to-Speech Synthesis. *Speech Communication* 48(6), 716–726 (2006)
11. Chu, M., Peng, H.: An Objective Measure for Estimating MOS of Synthesized Speech. In: Proc. of Eurospeech-2001, Aalborg, pp. 2087–2090 (2001)
12. Chu, M., Zhao, Y., Chen, Y.N., Wang, L.J., Soong, F.: The Paradigm for Creating Multi-Lingual Text-to-Speech Voice Database. In: Huo, Q., Ma, B., Chng, E.-S., Li, H. (eds.) ISCSLP 2006. LNCS (LNAI), vol. 4274, pp. 736–747. Springer, Heidelberg (2006)
13. McAulay, R.J., Quatieri, T.F.: Speech Analysis/Synthesis Based on a Sinusoidal Representation. *IEEE Trans. ASSP-34*(4), 744–754 (1986)