An ARM-based Embedded System Design for Speech-to-Speech Translation

Shun-Chieh Lin, Jhing-Fa Wang, Jia-Ching Wang, and Hsueh-Wei Yang

Department of Electrical Engineering, National Cheng Kung University,
No.1, Dasyue Rd., East District,
Tainan City, 70101, Taiwan, R.O.C.
{jason, wangjf, wjc, thyndo}@icwang.ee.ncku.edu.tw

Abstract. Previous research shows that there are two architectures for speech-to-speech translation (S2ST) system implementation. One is client-server based systems that should be built on the server computer but not available anytime or anywhere. The other is to build portable stand-alone devices but lacks the real-time performance. Therefore, this work presents an embedded system design for portable S2ST applications. This system is characterized by small size, low cost, real-time operation, and high portability. For realization of the proposed S2ST system, this work designs the ARM-based SoPC architecture, the speech translation intellectual property, and software procedures of the proposed SoPC. The entire design was implemented on ALTERA EPXA10. The English-to-Mandarin translation process can be completed within 0.5 second at a 40 MHz clock frequency with 1,200 translation patterns. The maximum frequency is 46.22 MHz, and the usage of logic elements is 19,318 (50% of the total logic elements of the EPXA10 device).

1 Introduction

Speech-to-speech translation (S2ST) has made significant advances over the past decade, with several high-visibility projects (JANUS [1], Verbmobil [2], EUTRANS [4], MATRIX [5], and others [3], [6], [7]) significantly advancing the state-of-the-art. Table 1 shows a comparison among the related S2ST systems. There are two architectures for speech-to-speech translation system implementation – server-client based systems and stand-alone handheld devices. For server-client based systems, a system is available on a central S2ST server and client access of the S2ST server is made possible by using digital cellular phones and internet. For S2ST handheld devices, they enable foreign tourists and business travelers to talk to locals through them for face-to-face communication.

Nevertheless, a critical shortcoming of server-client based S2ST systems is that they should be built on the server computer. In other words, while taking face-to-face cross-language communication to locals, it is not available anytime or anywhere. An obvious solution would be to build portable stand-alone S2ST devices. However, previous works show that real-time S2ST with a resource-limited device is still a problem. Moreover, both of these two architectures are platform-dependent S2ST

applications, i.e., they need some kinds of operation systems. This limitation would increase cost and result in lower productivity and embedability in other information appliances (IA) for cross-language voice communication.

Table 1. A Comparison among Related Spoken Language Translation Systems

System	Vocabulary Size	Response time	Specification
JANUS [1]	3,000~5,000	≤ 2 times real-time	PC-based platform (server-client)
Verbmobil [2]	≥10,000	4 times real-time	PC-based platform (server-client)
EUTRANS [4]	1,701(English) 2,459(Italian)	≤ 3 times real-time	PC-based platform (server-client)
MATRIX [5]	13,000	0.1 sec for TDMT	PC-based platform (server-client)
Isotani *et al.* [7]	20,000(English) 50,000(Japan)	Slower than V_R5500 processor	Pocket PC PDA (206 MHz StrongARM/64 MB RAM)
Speechalator [8]	–	≥2~3 sec	Pocket PC PDA (400 MHz StrongARM/64 MB RAM)

Therefore, an embedded system design solution is proposed by realizing the entire S2ST system within a single chip. This chip only requires a few peripheral components for complete operation, and is characterized by small size, low cost, real-time operation, high productivity and embedability for any information appliance (IA). The entire proposed S2ST system is implemented with ARM-based system-on-a-programmable-chip (SoPC) platform –Altera™ Excalibur™ EPXA10 development board. The rest of this paper is organized as follows. Section 2 gives the framework of the proposed S2ST embedded system. Section 3 discusses the ARM-based hardware/software co-design of the SoPC architecture. The implementation result is given in Section 4. Finally, conclusions and future works are provided in Section 5.

2 Framework of the Proposed Embedded System

2.1 The Proposed S2ST Process

The proposed speech-to-speech translation is based on a two-layer translation template retrieval method, a kind of integration of speech analysis and language translation [8]. The adopted template retrieval approach is directly from speech to speech without language models like other automatic speech recognition (ASR) approaches. With identifying speech features based on one-stage algorithm [9], translation primarily stays in the speech modality and does not go through a textual modality. The adopted method not only retrieves the optimal translation template, but also extracts the appropriate target patterns.

To formulate the template retrieval between input speech (X_1^L) and the *v*-th translation template (r_v), we use the following notations: l represents the frame index within X_1^L, $1 \le l \le L$, j represents the translation pair $\langle s_j^v, t_j^v \rangle$ index within r_v,

$1 \le j \le J$, and k represents the frame index within the j-th source speech pattern s_j^v and its mapped target speech pattern denoted by t_j^v, $1 \le k \le K_j$. Then for each input frame, the one-stage based accumulated distortion $d_A(l, k, j)$ is defined by:

$$d_A(l,k,j) = d(l,k,j) + \min_{k-2 \le m \le k} \left(d_A(l-1,m,j)\right), \tag{1}$$

for $2 \le k \le K_j$, $1 \le j \le J$, where $d(l, k, j)$ is the local distortion between the l-th frame of X_1^L and the k-th frame of the source speech pattern s_j^v. The recursion in (1) is carried out for the internal frames (i.e., $k \ge 2$) of each source pattern. At the pattern boundary, i.e., when $k = 1$, the recursion can be calculated as:

$$d_A(l,1,j) = d(l,1,j) + \min\left[\min_{1 \le m \le J}\left(d_A(l-1,K_m,m)\right), d_A(l-1,1,j)\right]. \tag{2}$$

And the best path is determined by

$$d_G^v = \min_{1 \le j \le J}\left[d_A\left(L, K_j, j\right)\right]. \tag{3}$$

In this work, the retrieved template is decided by

$$\hat{v} = \arg\min_{1 \le v \le V}\left[d_G^v\right]. \tag{4}$$

According to the decided $\hat{v}$-th template from (4), the target speech patterns $\left\{t_j^{\hat{v}}\right\}_{j=1}^{J}$ can be obtained from $\left\{s_j^{\hat{v}}\right\}_{j=1}^{J}$, where $s_j^{\hat{v}}$ is identified and extracted. With the determined speech patterns, these waveforms are reordered and rearranged with adequate overlapping portions to generate speech with the waveform similarity overlap and add (WSOLA) algorithm [10]. In general, while inputting a source speech, the speech is adjusted by a signal pre-processing and used to extract speech features. The extracted features of the source speech are imported to identify speech input for each template features in template retrieval. A hypothesized template is then selected using a ranking process and the identified source patterns are extracted from it. According to the waveform translation database, the target patterns are obtained from the identified source patterns. The hypothesized target patterns are used to produce speech output via the corresponding speech generation template by waveform replacement based on the WSOLA method.

2.2 The Proposed SoPC Architecture

According to the computation complexity analysis of the S2ST system, the highest computational load comes from the template retrieval. Furthermore, while template size increases, more computational load raises for template retrieval. Therefore, the template retrieval procedure needs to be partitioned into hardware implementation. The remainder procedures of the proposed S2ST system is implemented by ARM assembler or C language and executed by ARM-based processor.

The timing scheduling diagram after the partitioning is illustrated in Fig. 1. While the system receives enough speech data from speech recording process, the feature extraction procedure is loaded after speech pre-processing and the extracted features are used to process template retrieval for each template and extract the identified patterns. After ranking for selecting a hypothesized template, the identified source patterns are used to translate into target waveform patterns for target speech generation and the generated target speech is outputted for users by speech playing process.

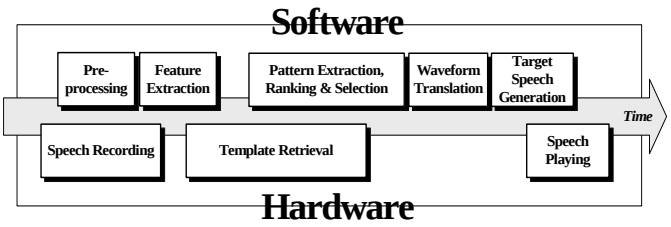

Fig. 1. Hardware/software scheduling after partitioning

According to the prior HW/SW partition results, besides the ARM-based processor, which is used for system control and software process, two specific intellectual properties (IPs) are designed for template retrieval and the controller of ADC/DAC circuit board. Figure 2 shows an overall block diagram of the SoPC architecture for the proposed S2ST system. The on-chip bus architecture of the proposed SoPC is based on the advanced microcontroller bus architecture (AMBA^{TM}). The proposed architecture contains the advanced high-performance bus (AHB) and the advanced peripheral bus (APB). ARM-based processor stripe is the only master of the AHB. The template retrieval IP and the AHB-APB Bridge are both AHB-compliant slave IP, and the controller for the converter board is an APB-compliant slave IP. Three major databases are stored in a FLASH memory. Software instruction codes and temporary data for ARM-based processor and template retrieval IP are stored in on-chip single-port SRAM and dual-port SRAM.

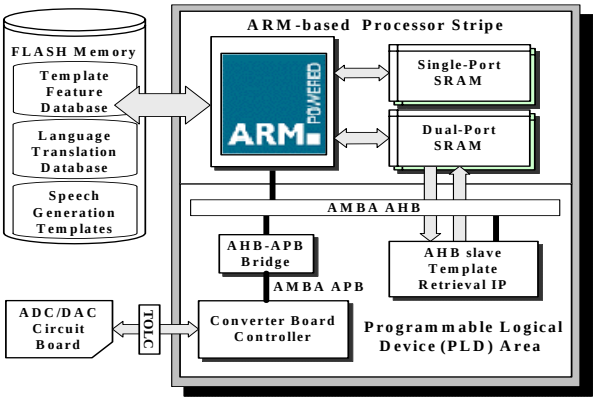

Fig. 2. SoPC architecture for the proposed S2ST system

3 ARM-based Hardware/Software Co-design

3.1 AHB Slave IP Design of Template Retrieval

There are four on-chip RAMs are used by the IP, as listed in Table 2. The template header RAM and the test feature RAM are wrapped in the AHB slave. ARM-based processor must write the speech input features and template header information to input feature RAM and template header RAM, respectively, via the AHB bus before validating template retrieval IP. The core of template retrieval IP is divided into two parts, the controller and the body. The function of controller is used to handle the operation of template retrieval IP by referring to the information which stored in the template header RAM. The AHB slave interface of template retrieval IP supports single and incrementing burst AHB transfers without read transfer.

Table 2. Description of RAMs for AHB-Compliant Template Retrieval IP

RAM	Memory Type	Size
Test feature RAM	Dual-port RAM built in PLD	8 Kbytes
Template header RAM	Dual-port RAM built in PLD	2 Kbytes
Template feature RAM	Dual-port SRAM 0 in ARM stripe	64 Kbytes
Decision RAM	Dual-port SRAM 1 in ARM stripe	64 Kbytes

Dependence Graph Projection of Template Retrieval. To exemplify our design concept, Figs. 3(a) and 3(b) display two dependence graphs (DGs) of the template retrieval. These dependence graphs are constructed according to (1) and (2), and can be regarded as the template retrieval space.

The DG shown in Fig. 3(a) describes the case with 12 frames in the input speech. There are three source patterns in this template, and the frame number of them is 7, 8, and 6, respectively. The DG shown in Fig. 3(b) has 8 frames in the input speech and 2 patterns in the template. The frame number of the two patterns in this template is 6 and 7, respectively. The different frame numbers within different input speeches cause the variation along the horizontal axis in the template retrieval space. The different pattern numbers within different templates and the different frame numbers within different patterns are responsible for the variation along the vertical axis in the template retrieval space. The architecture design for coping with the variation of the structure of the template retrieval space is described as follows. First, horizontal projection is performed in the original DGs (Fig. 3(a)) and the result is shown in Fig. 3(c). This projection reduces the two-dimensional (2-D) template retrieval space into a one-dimensional (1-D) one and eliminates the variation resulting from different frame numbers within different input speeches. Each node in Fig. 3(c) requires a register to store the computational results for the distortion accumulation of the next frame. The nodes within one column are divided into different blocks according to the patterns they belong to. Thus the DG in Fig. 3(c) consists of three blocks corresponding to three different patterns.

To further reduce the size of the DG, a vertical projection is performed in Fig. 3(c). This projection transforms the three-block DG into a one-block DG (see Fig. 3(e)). Because the frame numbers for the blocks are different, the largest frame number is

taken as the frame number for the new one-block DG. To deal with the discrepancy of having different frame numbers within different patterns, we added some multiplexers. In addition, two extra registers are added in front of each node to replace the two eliminated blocks. Figure 3(e) is then modified as Fig. 3(g) to have a regular wire connection. Figures 3(b), 3(d), 3(f) and 3(h) provide another example of the use of horizontal and vertical projections. One can find that the frame number and the register number for a certain node between the DGs in Figs. 3(g) and 3(h) are different. To combine the two DGs into a single one, the largest frame number and the largest register number between them are chosen, see Fig. 3(i). The added multiplexers in front of each node provide the selectivity among the one-block DGs.

Hardware Design of Template Retrieval

Distortion Unit Design. Each node in the template retrieval DG associates with a local distortion. The local distortion between a frame of the template and a frame of the input speech is defined by

$$\sum_{i=0}^{P} \left| R_i - U_i \right|, \tag{5}$$

where R_i and U_i denote the *i*-th cepstral coefficient in the *R*-th frame of the template and the *U*-th frame of the input speech, respectively, and *P* is the cepstrum order.

SIPO Unit Design. The serial-in parallel-out unit acts as an interface between the distortion unit and all the PEs. The local distortions obtained from the distortion unit are serially fed into the SIPO unit. After all local distortions for a block have been put into the SIPO register, they are sent to all the PEs in parallel.

PE Unit Design. Each node in the DGs shown in Fig. 3(i) is implemented as a processing element. After receiving all the accumulated distortions from the registers, the PE selects the minimum accumulated distortion, and then adds it to the local distortion $d(l,k,j)$. The result is the new accumulated distortion $d_A(l,k,j)$ associated with the current node. In addition to the accumulation process, the PE also generates the decision information that indicates the source node in a path transition.

Integrated Architecture. According to the situation of a general conversation, the architecture of Fig. 3 has been modified to accommodate 16 patterns for each template and 31 frames for each pattern. Assume that the template number of the proposed S2ST system is 100, the average amount of pattern for each template is 12, and the average frame number for each pattern is 25. A user input a series of speech signal for 3 seconds with 8 KHz sampling rate i.e. the input contains about 126 frames. The required clock cycles for the process of template retrieval IP is 18.9 M $(126 \times 12 \times 25 \times 100 \times 10 \div 2)$. If the clock frequency is 40 MHz(i.e. clock cycle = 25ns), the processing time is 0.4725 second $(18.9 \times 10^6 \times 25 \times 10^{-9})$, which meets real-time constrains by comparing with Table 1.

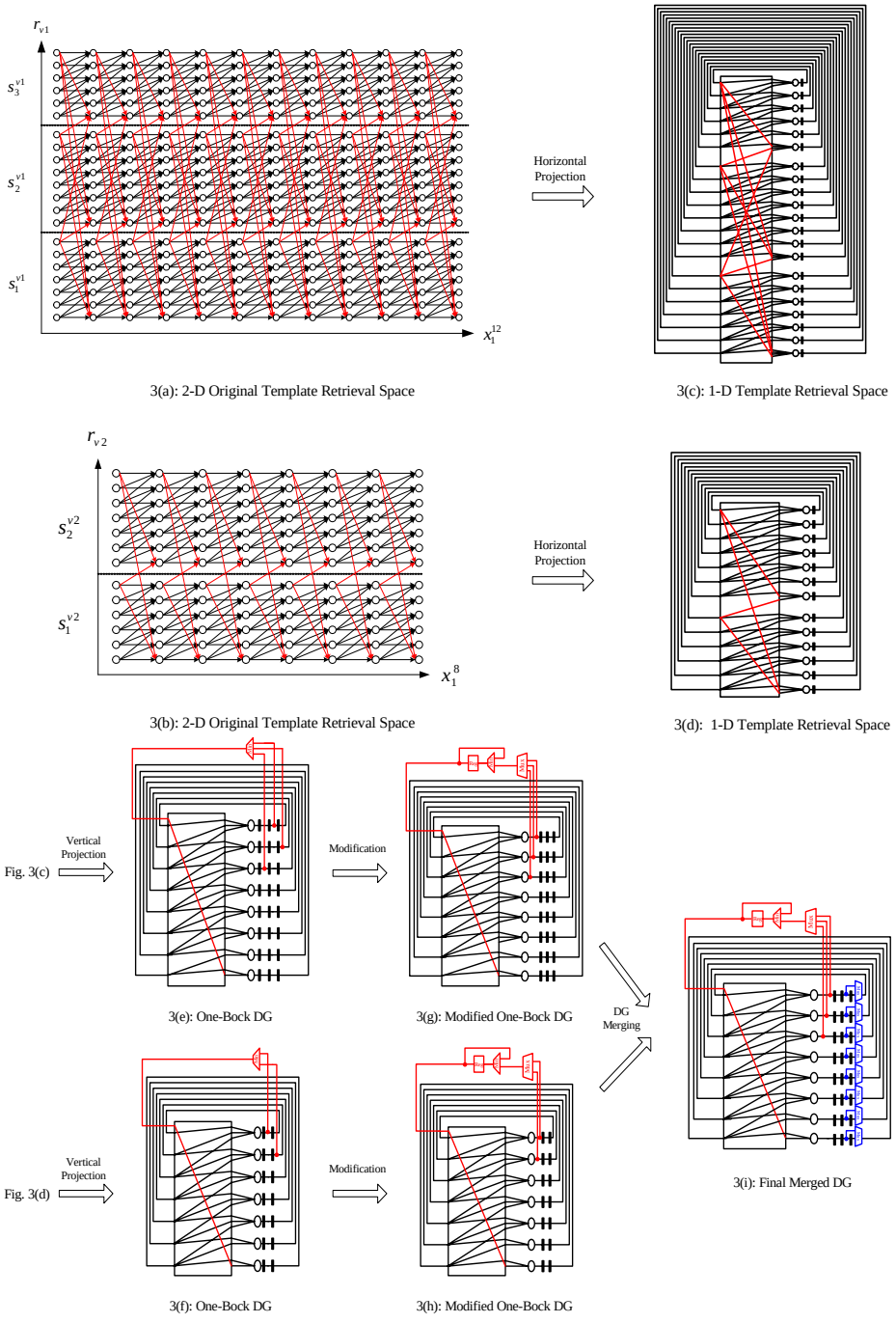

Fig. 3. Architecture inference of the template retrieval based on DG

3.2 Memory Allocation

There are two 128 Kbytes single-port SRAMs and two 64 Kbytes dual-port SRAMs within the embedded stripe of Excalibur device. The software instruction code and temporary data which are executed by ARM-based processor are stored in the single-port SRAMs. The two dual-port SRAMs are reserved for template features and decision path information of template retrieval. The 16 Mbytes FLASH memory is used to store the S2ST databases including template feature database, waveform translation database and speech generation templates. At the system startup, the template header block and the template feature block of template feature database are copied to the template header RAM of template retrieval IP and the dual-port SRAM 0 by ARM-based processor, respectively.

3.3 Software Procedure

Preprocessing of Speech Signal. To improve the performance on speech translation, the pre-processing for the proposed system contains three steps. The first pre-processing step is to calculate the offset for DC shifting automatically. Second, the energy-based approach [11], which is a traditional approach and works well under high SNR conditions, is applied here to eliminate silence components and avoid noise component generation. The finally steps is removing the redundant silence segments.

Pattern Extraction Procedure. To extract the best pattern sequence, we adopt a block-node pattern extraction method. Figure 4 exemplifies the relationship between nodes and blocks. The block boundaries are the same as the pattern boundaries. This template retrieval space contains 24 blocks, numbered from 0 to 23. Blocks in the same row belong to the same pattern, and have the same number of nodes (frames). For example, blocks 2, 5, 8, …, 23 belong to the third pattern within this template. The pattern extraction method can be regarded as a two-level address decoding process. At the first level, the blocks are the addressing units. Each block consists of frame nodes, which become addressing units at the second level. If b denotes the block index and k denotes the local node index in a block, each node can also be referred to by the pair (b, k). For example, node 64 can be referred to as node (21, 1).

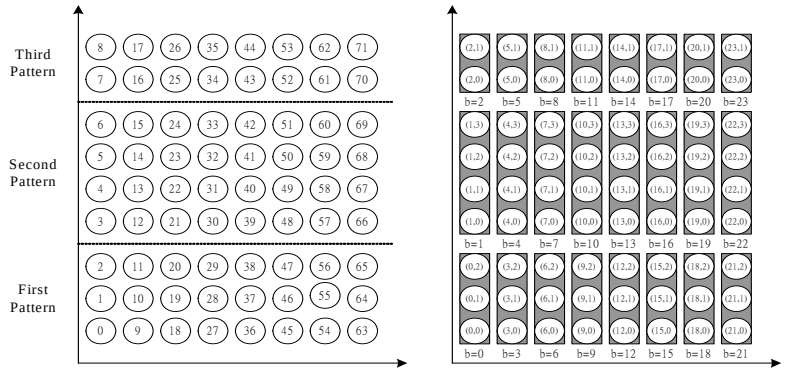

Fig. 4. Example illustrating the block-node addressing method

The decision word is generated for each block, and is denoted by $D \equiv \{d_0, d_1, d_2, \cdots, d_{K-1}, e\}$, where K is the largest number among all the block node ones, e is the pattern decision information from the first node of a block, and d_k, $0 \le k \le K-1$, is the decision information from the k-th node of a block. We use e to indicate the source pattern for an external (between-patterns) transition, and d_k to indicate the source node for an internal (within-patterns) transition. Assume that K_j denotes the number of nodes in the j-th pattern. Table 3 lists the transition information described by the decision word (K = 31 and *max_pattern_number* = 16, in the proposed S2ST system).

Table 3. Transition Information and Bit Allocation of a Decision Word

Decision information	Transition information	Bit allocation
d_0	d_0= 0: internal transition, i.e. (l-1, k, j)→(l, k, j);	1 bit
	d_0= 1: external transition, referred to e	
d_1	d_1= 0: (l-1, k, j)→ (l, k, j);	1 bit
	d_1= 1: (l-1, k-1, j)→ (l, k, j);	
$d_2, \cdots, d_k, \cdots, d_{K-1}$	d_k = 0: (l-1, k, j)→ (l, k, j);	2 bits
	d_k = 1: (l-1, k-1, j)→ (l, k, j);	
	d_k = 2: (l-1, k-2, j)→ (l, k, j);	
e	(l-1, K_e-1, e)→ (l, k, j);	$\lceil \log_2 max\_pattern\_number \rceil$ bits

4 Implementation Results

The user-interfaces on EPXA10 are four push-button switches, eight LEDs and a bit of DIP switch. The four push-button switches and the bit of DIP switch connect to the five individual interrupt signals to the ARM-based processor. Resource summary of the proposed system on EPXA10 are listed in Table 4. The maximum frequency is 46.22 MHz, and the usage of logic elements is 19,318 (50% of the total logic elements of the EPXA10).

5 Conclusions and Future Works

For a portable and real-time speech-to-speech translation embedded system, this work proposes the system-on-a-programmable chip (SoPC) realization. The system design is implemented with ARM-based Altera EPXA10 SoPC development board. The speech translation process can be completed within 0.5 second at a 40 MHz clock rate with 1,200 translation patterns while the number of templates is 100. The prototype chip was tested and found to be capable of performing within the expected specifications of real time. In the future, with various design of software or firmware, the proposed SoPC architecture would be used for different applications of pattern recognition. In addition, the AMBA-based architecture of the proposed S2ST SoPC would effortlessly integrate with other AMBA-compliant hardware devices for embedding and providing the function of speech translation on them.

Table 4. Summary of Resource Usage on EPXA10

Resource	Usage
Device	EPXA10F1020C2
Logic cells	19318 / 38400 (50 %)
Registers	9480/ 42658 (22 %)
I/O pins	79 / 715 (10 %)
Global signals	5
ESBs	40 / 160 (25 %)
Macrocells	0 / 2560 (0 %)
ESB pterm / CAM bits used	0 / 327680 (0 %)
Total memory bits	81920 / 327680 (25 %)
FastRow interconnects	0 / 120 (0 %)
PLLs	0 / 4 (0 %)
LVDS transmitters & LVDS receivers	0 / 16 (0 %)
Maximum fan-out node	hclock
Maximum fan-out	9509
Total fan-out	77777
Average fan-out	4

References

1. Lavie, A., Waibel, A., Levin, L., Finke, M., Gates, D., Gavaldá, M., Zeppenfeld, T., Zhan, P.: JANUS III: Speech-to-Speech Translation in Multiple Languages. Proc. IEEE Int. Conf. Acous., Speech, & Sig. Processing (1997) 99–102
2. Wahlster, W. (ed.): Verbmobil: Foundations of Speech-to-Speech Translation. Springer-Verlag, Berlin Heidelberg New York (2000)
3. Ney, H., Nießen, S., F. Och, J., Sawaf, H., Tillmann, C., Vogel, S.: Algorithms for Statistical Translation of Spoken Language. IEEE Trans. Speech Audio Processing 8 (2000) 24–36
4. Casacuberta, F., Llorens, D., Martnez, C., Molau, S., Nevado, F., Ney, H., Pastor, M.. Pico, D., Sanchis, A., Vidal, E., Vilar, J. M.: Speech-to-Speech Translation based on Finite-State Transducers. Proc. IEEE Int. Conf. Acous., Speech, & Sig. Processing (2001) 613–616
5. Sugaya, F., Takezawa, T., Yokoo, A., Yamamoto, S.: End-to-End Evaluation in ATR-MATRIX: Speech Translation System between English and Japanese. Proc. Eur. Conf. Speech Comm. Tech. (1999) 2431–2434
6. Isotani, R., Yamabana, K., Ando, S., Hanazawa, K., Ishikawa, S., Emori, T., Hattori, H., Okumura, A., Watanabe, T.: An Automatic Speech Translation System on PDAs for Travel Conversation. Proc. IEEE Int. Conf. Multimodal Interfaces (2002) 211–216
7. Waibel, A., Badran, A., Black, A. W., Frederking, R., Gates, D., Lavie, A., Levin, L., Lenzo, K., Tomokiyo, L. M., Reichert, J., Schultz, T., Wallace, D., Woszczyna, M., Zhang, J.: Speechalator: Two-Way Speech-to-Speech Translation on a Consumer PDA. Proc. Eur. Conf. Speech Comm. Tech. (2003) 369–372
8. Wang, J. F., Lin, S. C., Yang, H. W.: A New Two-Layer Approach for Speech-to-Speech Translation. Proc. Int. Sym. Chinese Spoken Language Processing (2004) 321–324
9. Rabiner, L. R., Juang, B. H. (ed.): Fundamentals of Speech Recognition. Prentice-Hall (1993)
10. Verhelst, W., Roelands, M.: An Overlap-Add Technique based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech. Proc. IEEE Int. Conf. Acous., Speech, & Sig. Processing (1993) 554–557
11. Rabiner, L. R., Sambru, M. R.: An Algorithm for Determining the Endpoints of Isolated Utterances. The Bell System Technical Journal 54 2 (1975) 297–315