Network and Embedded Applications of Automatic Speech Recognition

Nobuo Hataoka[1], **Hiroaki Kokubo**[2], **Akinobu Lee**[3],
Tatsuya Kawahara[4], and **Kiyohiro Shikano**[5], Non-members

ABSTRACT

ASR (Automatic Speech Recognition) is one of key technologies in the upcoming Ubiquitous Computing and Ambient Intelligence. In this paper, first, the surveys on processing devices such as microprocessors and memories, and on communication infrastructure, especially wireless communication infrastructure relating to ASR are reported. Second, the embedded version of CSR (Continuous Speech Recognition) software for the mobile environmental use of ASR is reported.

As the devices, RISC based microprocessors, semiconductor memories, and HDD are summarized. For the communication infrastructure, mobile communications and wireless LANs are described. Finally, implementation results of the free CSR software called Julius on the T-engineTM consisting of an SH-4A microprocessor are reported.

Keywords: Ubiquitous Computing, Ambient Intelligence, Automatic Speech Recognition (ASR), Continuous Speech Recognition (CSR), Julius: Free CSR Software, Embedded Julius, T-Engine, *SuperH* Microprocessor

1. INTRODUCTION

The Ubiquitous Computing was named and proposed by Mark Weiser[1]. Its concept contains "Anytime, Anywhere, Anybody," but now the philosophy expands to "this time, this place, this person." Recently, the Ambient Intelligence has been announced by Philips showing the Philipss R&D concept[2]. These two concepts indicate the same IT (Information Technology) world. Both need small devices, fast communication networks, RFID(Radio Frequency IDentification), and sophisticated human interfaces and terminals.

For Ubiquitous Computing and Ambient Intelligence, media processing technologies including speech are necessary to provide sophisticated human interfaces.

Concerning ASR, the CSR (Continuous Speech Recognition) software has been available on PCs (Personal Computers) which have huge computing resources, both computing power and memories. Our goal is to develop embedded CSR software which runs on small computing power and with small memory to extend ASR to mobile environmental use. We envision mobile application environments such as car navigation systems and cellular phones where an embedded speech recognizer[3] is running on connecting to remote servers via communication networks.

In this paper, the surveys on processing devices such as microprocessors and memories, and the surveys on communication infrastructure, especially wireless communication infrastructure are reported. These technologies are key factors to make Ubiquitous Computing and Ambient Intelligence possible. Finally, the embedded version of CSR (Continuous Speech Recognition) software is reported. We called this embedded CSR software the embedded version of Julius[4].

2. UBIQUITOUS COMPUTING AND AMBIENT INTELLIGENCE ENVIRONMENT

2.1 System Image for Networks and Terminals

As information technologies expand into the mobile environments to provide ubiquitous communication, an intelligent interface will be a key element to enable mobile access to networked information. For mobile information access, HMIs (Human Machine Interfaces) using speech might be the most important and essential because speech interfaces are more effective for small and portable devices. Mobile terminals such as cellular phones, PDAs (Personal Digital Assistants) and Hand-held PCs are already connected to networks such as internet to access information on web servers. Especially, Car Telematics refers to a new service concept where mobile terminals (e.g. car navigation systems, cellular phones) are used to connect to the information servers via communication networks[5].

As the speech processing environments, there are

Manuscript received on February 27, 2008 ; revised on , .

[1] The author is with Tohoku Institute of Technology, Sendai 982-8577, JAPAN, E-mail: hataoka@tohtech.ac.jp

[2] The author is with Central Research Laboratory, Hitachi Ltd, Kokubunji, Tokyo 185-8601, JAPAN, E-mail: hiroaki.kokubo.dz @hitachi.com

[3] The author is with Nagoya Institute of Technology, Nagoya, JAPAN, E-mail: ri@nitech.ac.jp

[4] The author is with Kyoto University, Kyoto, JAPAN, E-mail: kawahara@i.kyoto-u.ac.jp

[5] The author is with Nara Institute of Science and Technology (NAIST), Nara, JAPAN, E-mail: shikano@is.naist.jp

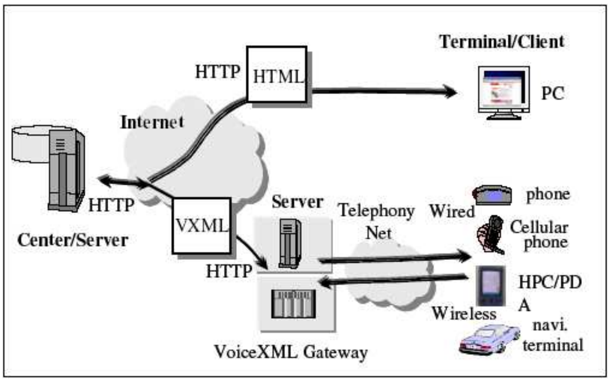
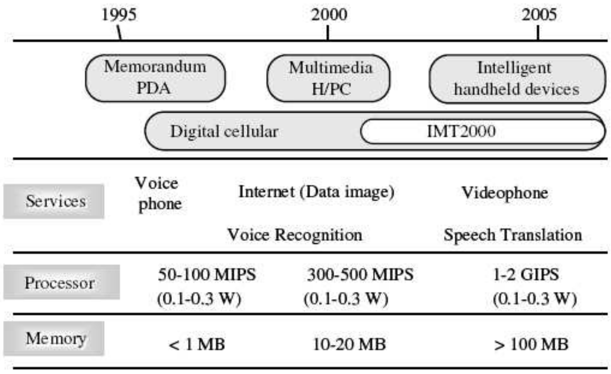


Fig.1: *System Image (Terminal, Network, and Center)*

hardware devices such as CPU and memory which speech media processing including speech recognition and speech synthesis runs on, and communication infrastructure to connect to application servers. Figure 1 shows a system image consisting of terminal/client, internet, and center/server. The processing devices are hardware-related devices such as PCs, microprocessors, and memories. The communication infrastructure includes wired and wireless environments.

2.2 Hardware Needs for Media Processing

Figure 2 summarizes hardware needs for mobile terminals such as HPC (Hand-held PC) and PDA. In the case of multi-language speech translation, CPU over 2 GIPS (Giga Instruction per Second)and memory size over 100MByte will be needed.



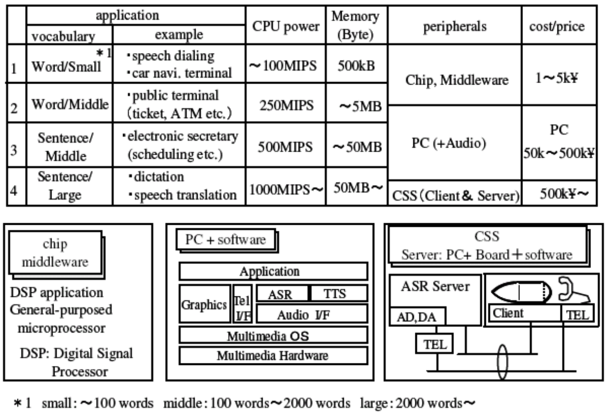
	1995	2000	2005
	Memorandum PDA	Multimedia H/PC	Intelligent handheld devices
	Digital cellular		IMT2000
Services	Voice phone	Internet (Data image)	Videophone
		Voice Recognition	Speech Translation
Processor	50-100 MIPS (0.1-0.3 W)	300-500 MIPS (0.1-0.3 W)	1-2 GIPS (0.1-0.3 W)
Memory	< 1 MB	10-20 MB	> 100 MB

Fig.2: *Hardware Needs for HPC/PDA Applications*

2.3 Implementation Varieties of ASR

Figure 3 shows implementation varieties of Automatic Speech Recognition (ASR) according to speech applications. Depending on the processing power and cost, there are three types of structures. The first one is chip and middleware on microprocessors. The second one is PCs and software implementation, and third one is a CSS (Client and Server System) structure.

To summarize, around 500MIPS CPU power and 50MByte memory size have been currently available one microprocessor and these hardware environments will make it possible to implement continuous speech recognition software on a microprocessor.



	application		CPU power	Memory (Byte)	peripherals	cost/price
	vocabulary	example				
1	Word/Small [*1]	・speech dialing ・car navi. terminal	～100MIPS	500kB	Chip, Middleware	1～5k¥
2	Word/Middle	・public terminal (ticket, ATM etc.)	250MIPS	～5MB		
3	Sentence/ Middle	・electronic secretary (scheduling etc.)	500MIPS	～50MB	PC (+Audio)	PC 50k～500k¥
4	Sentence/ Large	・dictation ・speech translation	1000MIPS～	50MB～	CSS (Client & Server)	500k¥～

*1 small: ～100 words middle: 100 words～2000 words large: 2000 words～

Fig.3: *Implementation Varieties of ASR*

3. DEVICE ENVIRONMENT

3.1 Microprocessor MPU

Due to improvements in microprocessor performance, various media processing technologies such as MPEG Codec and speech processing are possible to realize by software implementation. This technical trend depends not only on device progress but also on algorithm progress and progress of developmental/tool environments.

Figure 4 shows microprocessor product trend after the first product of 4004 was released. We watch on the microprocessors for digital consumer product use.

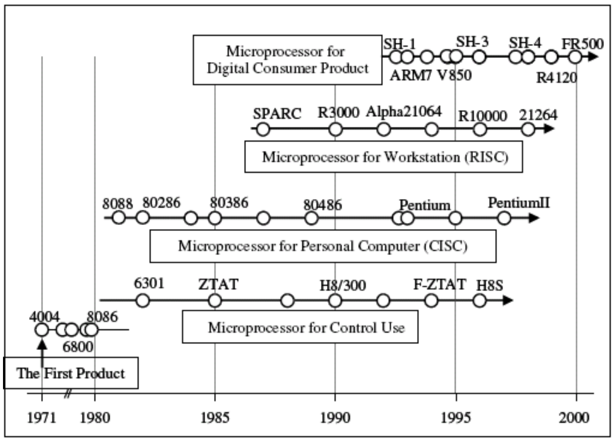


Fig.4: *Microprocessor Product Trend*

Figure 5 shows a road map for microprocessor, especially SuperH series. The SuperH microprocessors are products of Renesas Tech. Corp. and Hitachi

Ltd. and there are 3 types depending on applications. The first type is for mobile communication using control MPU such as SH2. The second one is for HPC/cellular phone applications using SH3 series which are characterized by low power MPU. The third type is for multimedia applications using high performance MPU, SH4 series.

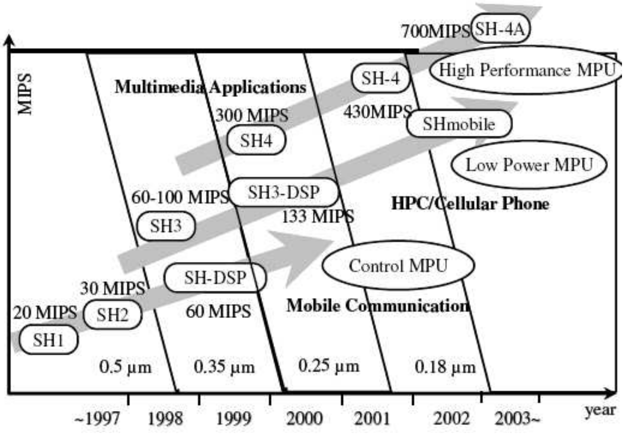


Fig.5: *Road Map for Microprocessors (SuperH Series)*

3.2 Memory

Two types of memories are surveyed, e.g. the semiconductor memory such as DRAM (Dynamic Random Access Memory) and flash memory, and HDD (Hard Disc Drive). Currently, DRAM memory size becomes over 8GByte because of the process improvement to less than 100nm (0.1 μm). For the terminal use, flash memory will be used widely from now on.

For the car navigation systems, currently 2 or 3 GByte HDD of 2.5 inch size has been used. Recently, smaller size HDD such as around 1 inch with 4GByte has been available for music players/terminals.

3.3 Device related OS

T-engine$^{\mathrm{TM}}$[6] has been available as the hardware platform to the network consumer terminals at the Ubiquitous Information Era. T-engine runs using TRON OS (Operating System). μiTron is used widely at the car navigation terminals because of its real time processing ability.

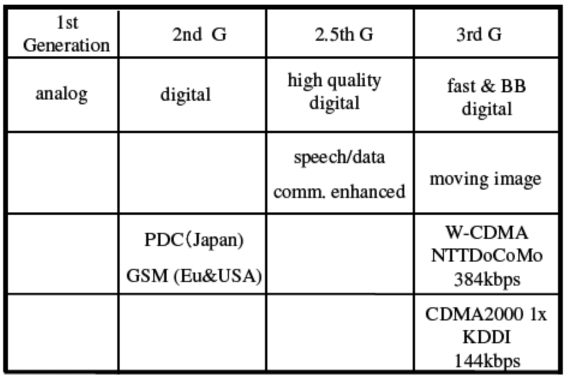
4. COMMUNICATION INFRA

4.1 Wireless Communication Infrastructure

In this wireless communication area, R&D competitions are active and the carriers activities become aggressive.

(1) Mobile Wireless
Figure 6 shows a cellular phone trend from the 1st generation to the 3rd generation. The 3rd generation is called IMT2000 also and this development has been organized according to the ITU (International Telecommunication Union) recommendations/guidelines. The ITU recommends international roaming and realization of 2Mbps communication speed/throughput.

The data throughput of the 2nd generation (PDC etc.) is 9.6kbps (upload) and 9.6kbps or 27.8kbps (download), and that of the 3rd generation such as FOMA (W-CDMA) is 64kbps (upload) and over 200kbps (download).



1st Generation	2nd G	2.5th G	3rd G
analog	digital	high quality digital	fast & BB digital
		speech/data comm. enhanced	moving image
	PDC(Japan) GSM (Eu&USA)		W-CDMA NTTDoCoMo 384kbps
			CDMA2000 1x KDDI 144kbps

Fig.6: *Cellular (Mobile/Portable) Phone Trend*

(2) High Speed Wireless LAN
Figure 7 shows the relationship among the standards of IEEE802.11a, 11b, 11g. The standard of 802.11b (max 11Mbps) and 11g (max 54Mbps) use 2.4GHz communication band and 802.11a (max 54Mbps) uses 5GHz band. Recently, products supporting 11g have been available.

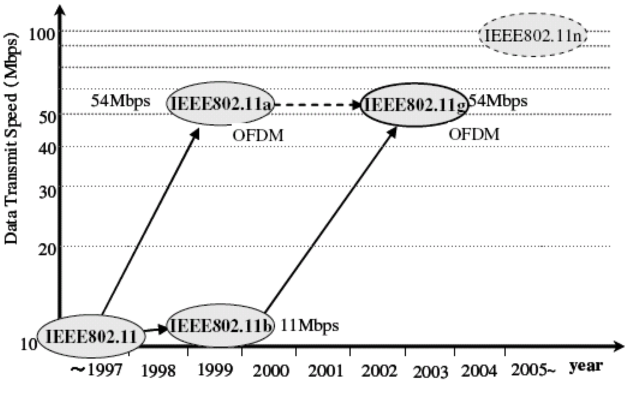


Fig.7: *Wireless Communication Technologies*

The 11a and 11g use OFDM (Orthogonal Frequency Division Multiplexing) as a modulation and coding method. OFDM technologies show significant quality at the single cells environments such as the surface-wave digital and wireless LAN. The development of MC-CDMA systems is going for the multi-cells and multi-users environments. The technical problems of

OFDM are high consuming power and multi-path error effects.

4.2 Speech Processing relating to Wireless Comm.

Distributed Speech Recognition (DSR) conducted by the AURORA project of The ETSI (European Telecommunications Standard Institute) has been proposed because of speech processing technologies for the 3rd generation communication infra. Speech analysis is done in terminals and the speech parameters are sent to centers to recognize the speech input (decoder).

5. EMBEDDED VERSION OF JULIUS

5.1 Free/Open CSR Software JULIUS

Julius is free and open CSR software which has been developed by Japanese Universities and delivered by WEB[7]. Julius can recognize large vocabulary over 20,000 words and running on Personal Computers (PCs) which have huge computing resources.

5.2 T-Engine with *SuperH* Microprocessor

T-Engine is a developmental hardware platform which has network security architecture and common Operating System (OS) called eTROM. The T-Engine board consists of a CPU board, an LCD board, and a debugging board. Figure 8 shows a photo of T-Engine and Table 1 shows T-Engine (MS7751RC01) specifications. We used Hitachis SuperH microprocessor called SH-4 which has 240 MHz/430MIPS CPU power on T-Engine.

Fig.8: *T-Engine Board*

The SH-4 is a RISC processor which has 32bit floating point calculation, and cache access commands. The work memory has 64MBytes, but only 55MBytes can be used for embedded software implementation. To implement Julius software on T-Engine, hardware modification was done for 16kHz sampling frequency and analog noise reduction.

5.3 Embedded CSR Implementation

Julius is free/open CSR software[7] which can recognize large vocabulary over 20,000 words, and running on PCs which have huge computing resources. Table 2 shows specifications of the embedded version of Julius implemented on T-Engine. Two conditions were checked to realize a real time processing. The first condition was a monophone type for acoustic models, and the second condition was a triphone type. For the language model, both conditions had bigram and trigram. The word accuracy rates on PCs were 86.05% and 90.655,000-word vocabulary size, respectively. Pre-evaluation for implementation of this Julius PC software on T-Engine showed far beyond the real time processing. Especially the initialization of acoustic models and program calling process needed over 10 minutes for 5,000-word vocabulary recognition.

Table 1: *T-Engine Specifications*

CPU	*SuperH* SH-4 (240MHz/430MIPS)
Flash Memory	8 MByte
Work Memory	64 MByte
OS	T-Kernel
Input/Output I/F	USB(Host), PCMCIA card, Serial, Headphone output, Microphone Input, LCD I/F, Extended buss I/F, etc.
LCD board	TFT color monitor 240×320
Size	120 mm×75 mm

Table 2: *Embedded Julius Specifications*

	Condition 1 (CND1)	Condition 2 (CND2)
Vocabulary size	5,000	←
Acoustic Models	Monophone	Triphone (PTM)
Language Models	bigram trigram	←
Beam width	400	←
Word accuracy (results on PC)	86.05%	90.65%

5.4 Implemental Issues

To realize the embedded version of Julius on T-Engine, the developmental issues are summarized as follows;

(1) CPU computing burden reduction:
The CPU power of T-Engine is restricted. Currently, the normal CPU power of PCs has been over 1.0GHz, however the T-Engine CPU power is around 200MHz.

Therefore, huge CPU burden reductions are necessary to realize real time processing on T-Engine.

(2) Memory burden reduction:
Especially, limitation of memory capacity on T-Engine is a fatal issue comparing to PCs which have over 1G Bytes memory size. Usually, 100M Bytes is a maximum memory size on embedded board environments. Therefore, huge memory reductions are needed for embedded use.

5.5 Preliminary Computing Process Reduction

(1)Compact Acoustic Models
The Julius is using an HTK format. The HTK format for acoustic models is Ascii type resulting that the model memory size is 12MByte. The binary encoding from Ascii encoding of acoustic models could lead huge memory size reduction from 12MByte to 3MByte.

(2)Addlog Table Memory Reduction
The addlog calculation was done before recognition process using a logarithm table. By this table memory assignment modification, huge memory reduction was done from 2MByte to 2kByte and no recognition accuracy distortion occurred.

(3)MFCC Calculation Speed-up
Speech parameter extraction of MFCC (Mel Frequency Cepstrum Coefficient) is reduced using cos. and sin. tables.

5.6 GMS Process Reduction

(1)GMS Calculation Burden
The GMS (Gaussian Mixture Selection) is a method to select Gaussian distributions by the HMM states using a hierarchical relationships between monophone models and triphone models[9]. This GMS method is introduced to reduce acoustic likelihood calculation, however the more process reduction is needed because the GMS calculation is almost half of the total process time.

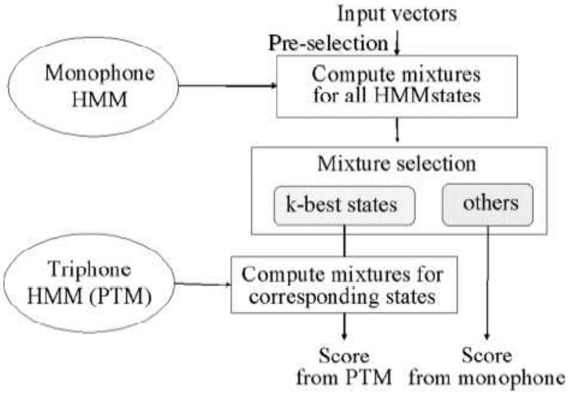


Fig.9: *GMS (Gaussian Mixture Selection)*

Figure 9 shows a GMS procedure. For each frame of input vectors, Gaussian mixtures for all monophone HMM states are computed and then Gaussian mixtures of triphone models are calculated for the only k-best states of monophone HMM models.

(2)Modifications on GMS
We made two modifications on the GMS method as follows;

(i)Computational Reduction on Pre-selection
The pruning process of a speech recognition decoder is done by the hypotheses that if scores are below a beam threshold, the calculations in pruned HMM states are not necessary. The only HMM states in active nodes need to be calculated.

In the conventional GMS, the scores of all monophone HMM states are calculated in a pre-selection stage. Knowing information of active nodes at the pre-selection stage, the only monophone HMM states linked to the active nodes can be calculated. Figure10 shows an image of the modified strategy. Filled circles are designated HMM states whose hypotheses stay alive, and unfilled circles are designated HMM states whose hypotheses are pruned. Figure 10(a) shows the conventional GMS. The pdf scores of all states are calculated and k-best states from among them are selected (meshed area). Possibly some states of k-best states have no active hypothesis. It is useless to calculate a pdf score of pruned hypothesis. Figure 10(b) shows the monophone models for the proposed and modified GMS. The target of pdf calculation is restricted to HMM states whose hypotheses stay alive (within a bold circle). Applying the modification, the computational cost is reduced for the mixture selection of the GMS compared to the conventional GMS. Furthermore, since there is no fear of selecting the useless states whose hypothesis is pruned, a small number of k-best could be specified without any degradation of recognition accuracy.

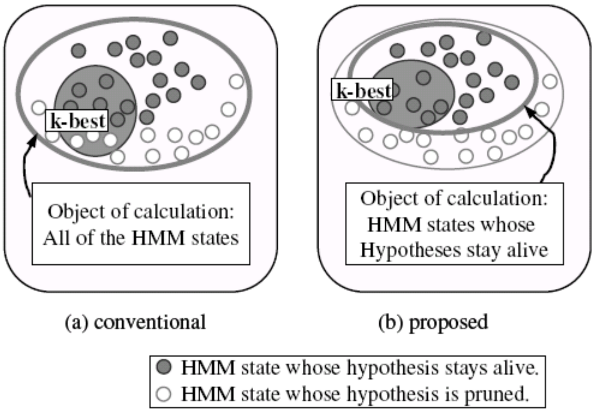

k-best
Object of calculation: All of the HMM states
k-best
Object of calculation: HMM states whose Hypotheses stay alive
(a) conventional
(b) proposed
● HMM state whose hypothesis stays alive.
○ HMM state whose hypothesis is pruned.

Fig.10: *Modification on Mixture Selection Stage*

(ii)Gaussian Selection within HMM State
The Gaussian Selection (GS)[10] is based on the idea that Score calculated by a Gaussian neighboring an input vector is dominant on score of HMM state. On the other hand, all Gaussians within HMM state are calculated in the original GMS, even though scores derived from Gaussians distant from input vectors are negligible.

For reducing calculations of scores on HMM states, we change calculation strategy: calculating only neighbor Gaussians of input vector, instead of calculating all Gaussians.

Figure 11 shows details of our strategy. The gmax is Gaussian maximum mixture score in an HMM state of monophone HMM. It is plausible idea that neighbor Gaussians of input vector in HMM state of triphone HMM are close to g_{max}. Based on this idea, only neighbors of g_{max} are calculated on HMM state, others, which are far from g_{max}, are omitted to calculate. By this procedure computation cost is much more reduced.

Distances between Gaussians can be calculated and stored in hash tables in advance.

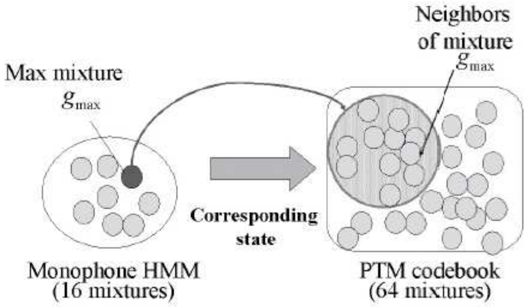

Neighbors of mixture g_{max}
Max mixture g_{max}
Corresponding state
Monophone HMM (16 mixtures)
PTM codebook (64 mixtures)

Fig.11: *GS (Gaussian Selection) in HMM state*

6. EVALUATION EXMERIMENTS

6.1 Experimental Setup

Table 3 shows experimental conditions for Julius software. The vocabulary size was 5,000 words and the triphone models had 3,000 states and 64 mixtures. The monophone models had 129 states and 16 mixtures.

Table 3: *Experimental setup for Julius*

vocabulary size	5,000 words
Triphone (PTM)	3,000 states, 64 mixtures
Monophone	129 states, 16 mixtures

6.2 Evaluation Results

(1)Preliminary Computing Process Reduction
There is no approximated process in this preliminary computing process reduction. This means no recognitionrate distortion occurs by this process reduction. However,the distortion by the board noise may occur, so we tested T-Engine performance evaluation first. For the T-Engine performance evaluation, we used line input from PC file speech to avoid utterance varieties and environmental noise varieties.

In details, the following procedures are used for the T-Engine board evaluation. First, the input speech is input to T-Engine from PC using a line input, and then the speech input is stored to the flash memory attached to T-Engine. This speech file is incorporated by the T-Engine internal noise. Utterances by 30 males and 30 female were used for the evaluation. Table 4 shows board evaluation results. In the table, the original shows results of file speech input meaning digitized data by PC. This means recognition results of original are the top recognition rates. Two conditions, monophone and triphone are set in the evaluation. The recognition rates of the condition 2 with triphone were 89.1% for original and 85.9% for T-Engine showing 5% recognition accuracy distortion by the T-Engine board.

Table 4: *Evaluation Results(1): word accuracy(ACC)*

	CND1: monophone		CND2: triphone	
	Original	T-Engine	Original	T-Engine
Male 30	78.2%	72.7%	86.7%	83.7%
Female 30	84.5%	79.7%	91.9%	88.2%
Total	81.3%	76.2%	89.1%	85.9%

(2) GMS Process Reduction
First, the recognition performance of the proposed GMS process reduction method was evaluated by the PC (Linux: Pentium4 2.8GHz) simulation. Figure 12 shows the evaluation results. Evaluation data were 100 sentences for each one male and one female from Japanese JNAS speech corpus. From the results, we found less k-neighbor value showed less word accuracy and the proposed GMS reduction method showed significant computing process time reduction (40% reduction) with small word accuracy loss (only 1%).

Next, the performance on the T-Engine (SH-4, 240MHz/430MIPS) platform was evaluated. The evaluation data were sentence utterances from 30 males and 30 females. The no. of k-neighbor was 24. Table 5 shows evaluation results on T-Engine. We found that requirement for the embedded Julius is less than 50MByte and that there was no big difference on word accuracy among no GMS, original GMS and the proposed GMS. The RTF (Real Time Factor) shows process length normalized by the utterance length. The proposed GMS showed 2.23 of RTF resulting 79% of that of no GMS. By the simulation, the process reduction by the proposed GMS was 40%. This difference may be occurred form T-Engine architecture and usage of cache memory.

Table 5: *Evaluation Results(2) on T-Engine*

	Memory size	word ACC	RTF
no GMS	48.9MBytes	89.1%	2.81
original GMS	49.8MBytes	89.3%	2.62
proposed GMS	49.8MBytes	89.7%	2.23

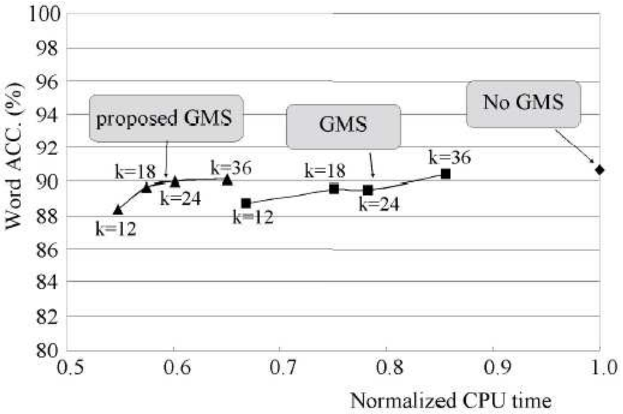


Fig.12: *Method Comparison: CPU time vs. Word ACC*

7. FUTURE WORK

We will investigate more compact and more noise robust embedded version of Julius which has 20,000-word vocabulary size[11]. The new CPU processor SH-4A (400MHz/700MIPS) will be used to get fast processing time. For the noise robustness, we are developing a new noise reduction process module at the front-end.

8. CONCLUSION

This paper describes surveys on processing devices and communication infrastructure, and reports the embedded version of Julius to provide sophisticated HMIs (Human Machine Interfaces). We used T-EngineTM (SH-4, 240MHz/430MIPS) as a hardware platform. We could realize 2.23 of RTF (Real Time Factor) of CSR processing on the condition of 5000-word vocabulary.

9. ACKNOWLEDGEMENT

The work has been supported by the e-Society Project founded by Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] Mark Weiser HP: http://www.ubiq.com/ubicomp/

[2] Philips: http://www.research.philips.com/technologies/

[3] N. Hataoka, et al. Proc. of IEEE ICASSP1998, pp.II837-II840, 1998.

[4] H. Kokubo, et al., "Embedded Julius: Continuous Speech Recognition Software for Microprocessor," in appearing *in Proc. of MMSP2006*, Canada, Oct., 2006.

[5] N. Hataoka, et al., "Robust Speech Dialog Interface for Car Telematics Service," *Proc of IEEE CCNC2004*, Las Vegas, Jan., 2004.

[6] T-Engine: http://www.t-engine.org/index.html

[7] Julius HP: http://julius.sourceforge.jp/en/julius.html/

[8] A. Lee, T. Kawahara, S. Doshita, "An Efficient Two-pass Search Algorithm using Word Trellis Index," *in Proc. of ICSLP*, pp.1831-1834, 1998.

[9] A. Lee and T. Kawahara and K. Shikano, "Gaussian Mixture Selection using Context-Independent HMM," *Proc. of IEEE ICASSP2001-1-18*, 2001.

[10] K. M. Knill, et al., "Use of Gaussian Selection in Large Vocabulary Continuous Speech Recognition using HMMs," in *Proc. of ICSLP*, vol. 1, pp. I-470-I-473, 1996.

[11] H. Kokubo, N. Hataoka, et al.,"Real-Time Continuous Speech Recognition System on SH-4A Micro- processor," *Proc. of MMSP2007*, Crete, Oct., 2007.

Nobuo Hataoka He received the B.S.E.E. degree and the M. Sc. degree in Electrical and Electronics Engineering from Tohoku University, in 1976 and 1978, respectively, and the Ph.D in Engineering in 1992 from Tohoku University. He joined Central Research Laboratory, Hitachi Ltd. in 1978, and he spent one year from 1988 to 1989 as Visiting Researcher at Carnegie Mellon University in U.S.A. From 1989 to 1993, he was Laboratory Manager of Hitachi Dublin Laboratory in Ireland and after returning to HCRL in Japan he took management responsibilities as Chief Research Scientist. He is currently Professor of Tohoku Institute of Technology in Sendai, Japan. His research interests include media implementation on microprocessors, and algorithm development on speech recognition , speech synthesis, speech translation, and artificial intelligence.

Dr. Hataoka is a member of the IEEE Acoustic, Speech, and Signal Processing Society, the Institute of Electronics, Information and Communication Engineers (IEICE), Japan, and the Acoustical Society of Japan. He is president in Information Systems Society of IEICE.

Hiroaki Kokubo received the B.S., M.S. and ph. D degrees from the science and engineering department in Sophia University, Tokyo, Japan in 1988, 1990, and 2003, respectively. From 1990, he joined Central Research Laboratories, Hitachi CO. LTD., Tokyo Japan. During 1995-1997, he was member of Interpreting Telecommunications Research Laboratories at ATR. From 2000 to 2004, he was member of Spoken Language Translation Research Laboratories at ATR. He is currently senior researcher of Central Research Laboratory, Hitachi CO. LTD. His research interest includes speech interface.

Akinobu Lee was born in Kyoto, Japan, on December 19, 1972. He received the B.E. and M.E. degrees in information science, and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 1996, 1998 and 2000, respectively. He worked on Nara Institute of Science and Technology as an assistance professor from 2000-2005. Currently he is an associate professor of Nagoya Institute of Technology, Japan. His research interests include large vocabulary continuous speech recognition, spoken language understanding, and real-world spoken dialogue system. He is a member of IEEE, ISCA, IPSJ and the Acoustical Society of Japan.

Tatsuya Kawahara received the B.E. degree in 1987, the M.E. degree in 1989, and the Ph.D. degree in 1995, all in information science, from Kyoto University, Kyoto, Japan. In 1990, he became a Research Associate in the Department of Information Science, Kyoto University. From 1995 to 1996, he was a Visiting Researcher at Bell Laboratories, Murray Hill, NJ, USA. Currently, he is a Professor in the Academic Center for Computing and Media Studies and an Adjunct Professor in the School of Informatics, Kyoto University. He is also an Invited Researcher at ATR Spoken Language Communication Research Laboratories.

He has published more than 150 technical papers covering speech recognition, spoken language processing, and spoken dialogue systems. He has been managing several speech-related projects in Japan including a free large vocabulary continuous speech recognition software project (http://julius.sourceforge.jp/).

Dr. Kawahara received the 1997 Awaya Memorial Award from the Acoustical Society of Japan and the 2000 Sakai Memorial Award from the Information Processing Society of Japan. From 2003 to 2006, he was a member of the IEEE SPS Speech Technical Committee. He was a general co-chair of IEEE Automatic Speech Recognition & Understanding workshop (ASRU-2007).

He is a member of the Acoustical Society of Japan (ASJ), the Institute of Electronics, Information and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Japanese Society of Artificial Intelligence (JSAI) and the IEEE.

Kiyohiro Shikano received the B.S., M.S., and Ph.D. degrees in electrical engineering from Nagoya University in 1970, 1972, and 1980, respectively. He is currently a professor of Nara Institute of Science and Technology (NAIST), where he is directing speech and acoustics laboratory. From 1972 to 1993, he had been working at NTT Laboratories. During 1986-1990, he was the Head of Speech Processing Department at ATR Interpreting Telephony Research Laboratories. During 1984-1986, he was a visiting scientist in Carnegie Mellon University. He received the Yonezawa Prize from IEICE in 1975, the Signal Processing Society 1990 Senior Award from IEEE in 1991, the Technical Development Award from ASJ in 1994, IPSJ Yamashita SIG Research Award in 2000, and Paper Award from the Virtual Reality Society of Japan in 2001, IEICE paper award in 2005 and 2006, and Inose award in 2005. He is a fellow of the Institute of Electrical and Electronics, Engineers (IEEE), the Institute of Electronics, Information and Communication Engineers of Japan (IEICE), and Information Processing Society of Japan, and a member of the Acoustical Society of Japan (ASJ), Japan VR Society, and International Speech Communication Society(ISCA).