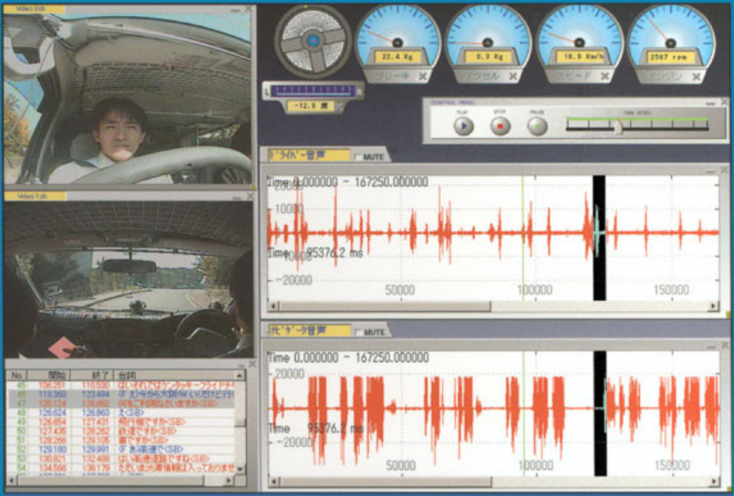
DSP for In-Vehicle and Mobile Systems



No.	開始	終了	台詞
45	106.251	110.530	はいそれではケンタッキーフライドチ
46	118.360	123.494	(F えっ)今から大阪のW(い)だけどすぐ
47	125.124	126.652	何をご利用なさいますか<SB>
48	126.624	126.863	え<SB>
49	126.654	127.431	飛行機ですか<SB>
50	127.435	128.262	鉄道ですか<SB>
51	128.266	129.105	車ですか<SB>
52	129.180	129.991	(F あ)高速で<SB>
53	130.821	132.468	はい高速道路ですね<SB>
54	134.566	136.179	ただいま渋滞情報は入っておりませ

Edited by

Hüseyin Abut

John H.L. Hansen

Kazuya Takeda

Springer

DSP FOR IN-VEHICLE AND MOBILE SYSTEMS

This page intentionally left blank

DSP FOR IN-VEHICLE AND MOBILE SYSTEMS

Edited by

Hüseyin Abut
Department of Electrical and Computer Engineering
San Diego State University, San Diego, California, USA
<abut@akhisar.sdsu.edu>

John H.L. Hansen
Robust Speech Processing Group, Center for Spoken Language Research
Dept. Speech, Language & Hearing Sciences, Dept. Electrical Engineering
University of Colorado, Boulder, Colorado, USA
<John.Hansen@colorado.edu>

Kazuya Takeda
Department of Media Science
Nagoya University, Nagoya, Japan
<takeda@is.nagoya-u.ac.jp>

Springer

eBook ISBN: 0-387-22979-5
Print ISBN: 0-387-22978-7

Created in the United States of America

Visit Springer's eBookstore at: http://ebooks.kluweronline.com
and the Springer Global Website Online at: http://www.springeronline.com

Dedication

To Professor Fumitada Itakura

This book, "DSP for In-Vehicle and Mobile Systems", contains a collection of research papers authored by prominent specialists in the field. It is dedicated to Professor Fumitada Itakura of Nagoya University. It is offered as a tribute to his sustained leadership in Digital Signal Processing during a professional career that spans both industry and academe. In many cases, the work reported in this volume has directly built upon or been influenced by the innovative genius of Professor Itakura.

While this outstanding book is a major contribution to our scientific literature, it represents but a small chapter in the anthology of technical contributions made by Professor Itakura. His purview has been broad. But always at the center has been digital signal theory, computational techniques, and human communication. In his early work, as a research scientist at the NTT Corporation, Itakura brought new thinking to bit-rate compression of speech signals. In partnership with Dr. S. Saito, he galvanized the attendees of the 1968 International Congress on Acoustics in Tokyo with his presentation of the Maximum Likelihood Method applied to analysis-synthesis telephony. The presentation included demonstration of speech transmission at 5400 bits/sec with quality higher than heretofore achieved. His concept of an all-pole recursive digital filter whose coefficients are constantly adapted to predict and match the short-time power spectrum of the speech signal caused many colleagues to hurry back to their labs and explore this new direction. From Itakura's stimulation flowed much new research that led to significant advances in linear prediction, the application of autocorrelation, and eventually useful links between cepstral coefficients and linear prediction. Itakura was active all along this route, contributing among other ideas, new knowledge about the Line Spectral Pair (LSP) as a robust means for encoding predictor coefficients. A valuable by-product of his notion of adaptively matching the power spectrum with an all-pole digital filter gave rise to the Itakura-Saito distance measure, later employed in speech recognition as well as a criterion for low-bit-rate coding, and also used extensively in evaluating speech enhancement algorithms.

Itakura's originality did not escape notice at Bell labs. After protracted legalities, a corporate arrangement was made for sustained exchange of research scientists between ATT and NTT. Fumitada Itakura was the first to initiate the program, which later encompassed such notables as Sadaoki Furui, Yoh`ichi Tohkura, Steve Levenson, David Roe, and subsequent others from both organizations. At Bell Labs during 1974 and -75, Fumitada ventured into automatic speech recognition, implementing an airline reservation system on an early laboratory computer. Upon his return to his home company Dr. Itakura was given new responsibilities in research management, and his personal reputation attracted exceptional engineering talent to his vibrant organization.

Following fifteen years of service with NTT, the challenges of academe beckoned, and Dr. Itakura was appointed Professor of Electrical Engineering in Nagoya University – the university which originally awarded his PhD degree. Since this time he has led research and education in Electrical Engineering, and Acoustic Signal Processing, all the while building upon his expertise in communications and computing. Sophisticated microphone systems to combat noise and reverberation were logical research targets, as exemplified by his paper with colleagues presented in this volume. And, he has continued management responsibilities in contributing to the leadership of the Nagoya University Center for Integrated Acoustic Information Research (CIAIR).

Throughout his professional career Professor Itakura has steadily garnered major recognition and technical awards, both national and international. But perhaps none rivals the gratification brought by the recognition bestowed by his own country in 2003 -- when in formal ceremony at the Imperial Palace, with his wife Nobuko in attendance, Professor Itakura was awarded the coveted Shiju-hosko Prize, also known as the Purple Ribbon Medal.

To his stellar record of career-long achievement we now add the dedication of this modest technical volume. Its pages are few by comparison to his accomplishments, but the book amply reflects the enormous regard in which Professor Fumitada Itakura is held by his colleagues around the world.

Jim Flanagan
Rutgers University

Table of Contents

List of Contributors

Hüseyin Abut, *San Diego State University, USA*
Hamid R. Abutalebi, *University of Yazd, Iran*
Ichiro Akahori, *Denso Corp., Japan*
Murat Akbacak, *University of Colorado at Boulder, USA*
Pongtep Angkititrakul, *University of Colorado at Boulder, USA*
Katsushi Asami, *Denso Corp., Japan*
Robert L. Brennan, *Dspfactory, Canada*
Alessio Brutti, *ITC-irst, Italy*
Guo Chen, Nanyang *Technological University, Singapore*
Chiasserini F. Chiasserini, *Politecnico di Torino, Italy*
Tan Eng Chong, *Nanyang Technological University, Singapore*
Paolo Coletti, *ITC-irst, Italy*
Luca Cristoforetti, *ITC-irst, Italy*
Juan Carlos De Martin, *Politecnico di Torino, Italy*
Michael Duggan, *Carnegie Mellon University, USA*
Engin Erzin, *Koç University, Turkey*
George H. Freeman, *University of Waterloo, Canada*
Sadaoki Furui, *Tokyo Institute of Technology, Japan*
Petra Geutner, *Robert Bosch, Germany*
Alessandro Giacomini, *ITC-irst, Italy*
Yuuichi Hamasuna, *DDS Inc., Japan*
John H.L. Hansen, *University of Colorado at Boulder, USA*
Masayasu Hata, *Chubu University, Japan*
Nobuo Hataoka, *Hitachi Ltd., Japan*
Diane Hirschfeld, *voice INTER connect, Germany*
Rüdiger Hoffmann, *Dresden University of Technology, Germany*
Kei Igarashi, *Nagoya University, Japan*
Yuki Irie, *Nagoya University, Japan*
Fumitada Itakura, *Nagoya University, Japan*
Koji Iwano, *Tokyo Institute of Technology, Japan*
Scott Judy, *Carnegie Mellon University, USA*
Shubha Kadambe, *HRL Laboratories, USA*
Nobuo Kawaguchi, *Nagoya University, Japan*
Itsuki Kishida, *Nagoya University, Japan*
Soo Ngee Koh, *Nanyang Technological University, Singapore*

List of Contributors (cont.)

Hiroshi Kondo, *Kyushu Institute of Technology, Japan*
Tahaharu Kouda, *Kyushu Institute of Technology, Japan*
Mirko Maistrello, *ITC-irst, Italy*
Enrico Masala, *Politecnico di Torino, Italy*
Marco Matassoni, *ITC-irst, Italy*
Shigeki Matsubara, *Nagoya University, Japan*
Michela Meo, *Politecnico di Torino, Italy*
Teruko Mitamura, *Carnegie Mellon University, USA*
Hiroya Murao, *Nagoya University, Japan*
Eric Nyberg, *Carnegie Mellon University, USA*
Yasunari Obuchi, *Hitachi Ltd., Japan*
Maurizio Omologo, *ITC-irst, Italy*
Bryan Pellom, *University of Colorado at Boulder, USA*
Rico Petrick, *voice INTER connect, Germany*
Thomas Richter, *voice INTER connect, Germany*
Takahiro Seki, *Tokyo Institute of Technology, Japan*
Antonio Servetti, *Politecnico di Torino, Italy*
Hamid Sheikhzadeh, *Dspfactory, Canada*
Teruo Shimomura, *Kyushu Institute of Technology, Japan*
Tetsuya Shinde, *Nagoya University, Japan*
Ing Yann Soon, *Nanyang Technological University, Singapore*
Frank Steffens, *Robert Bosch, Germany*
Piergiorgio Svaizer, *ITC-irst, Italy*
Kazuya Takeda, *Nagoya University, Japan*
Ichi Takumi, *Nagoya Institute of Technology, Japan*
Masahiko Tateishi, *Denso Corp., Japan*
A. Murat Tekalp, *Koç University, Turkey*
Abdul Wahab, *Nanyang Technological University, Singapore*
Hsien-chang Wang, *Taiwan, R.O.C.*
Jhing-fa Wang, *Taiwan, R.O.C.*
Wayne Ward, *University of Colorado at Boulder, USA*
Yukiko Yamaguchi, *Nagoya University, Japan*
Umit H. Yapanel, *University of Colorado at Boulder, USA*
Yücel Yemez, *Koç University, Turkey*
Xianxian Zhang, *University of Colorado at Boulder, USA*
Lifeng Zhang, *Kyushu Institute of Technology, Japan*

Preface

Over the past thirty years, much progress has been made in the field of automatic speech recognition (ASR). Research has progressed from basic recognition tasks involving digit strings in clean environments to more demanding and complex tasks involving large vocabulary continuous speech recognition. Yet, limits exist in the ability of these speech recognition systems to perform in real-world settings. Factors such as environmental noise, changes in acoustic or microphone conditions, variation in speaker and speaking style all significantly impact speech recognition performance for today systems. Yet, while speech recognition algorithm development has progressed, so has the need to transition these working platforms to real-world applications. It is expected that ASR will dominate the human-computer interface for the next generation in ubiquitous computing and information access. Mobile devices such as PDAs and cellular telephones are rapidly morphing into handheld communicators that provide universal access to information sources on the web, as well as supporting voice, image, and video communications. Voice and information portals on the WWW are rapidly expanding, and the need to provide user access to larger amounts of audio, speech, text, and image information is ever expanding. The vehicle represents one significant emerging domain where information access and integration is rapidly advancing. This textbook is focused on digital signal processing strategies for improving information access, command and control, and communications for in-vehicle environments. It is expected that the next generation of human-to-vehicle interfaces will incorporate speech, video/image, and wireless communication modalities to provide more efficient and safe operations within car environments. It is also expected that vehicles will become "smart" and provide a level of wireless information sharing of resources regarding road, weather, traffic, and other information that drivers may need immediately or request at a later time while driving on the road. It is also important to note that while human interface technology continues to evolve and expand, the demands placed on the vehicle operator must also be kept in mind to minimize task demands and increase safety.

The motivation for this textbook evolved from many high quality papers that were presented at the DSP in Mobile and Vehicular Systems Workshop, Nagoya, Japan, April 2003, with generous support from CIAIR, Nagoya University. From that workshop, a number of presentations were selected to be expanded for this textbook. The format of the textbook is centered about three themes: (i) in-vehicle corpora, (ii) speech recognition/dialog systems with emphasis on car environments, and (iii) DSP for mobile platforms involving noise suppression, image/video processing, and alternative communication scenarios that can be employed for in-vehicle applications.

The textbook begins with a discussion of speech corpora and systems for in-vehicle applications. Chapter 1 discusses a multiple level audio/video/data corpus for in-car dialog applications. Chapter 2 presents the CU-Move in-vehicle corpus, and an overview of the CU-Move in-vehicle system that includes microphone array processing, environmental sniffing, speech features and robust recognition, and route dialog navigation information server. Chapter 3 also focuses on corpus development, with a study on dialog management involving traffic, tourist, and restaurant information. Chapter 4 considers in-vehicle dialog scenario where more than one user is involved in the dialog task. Chapter 5 considers distributed task management for car telematics with emphasis on VoiceXML. Chapter 6 develops an in-vehicle voice interaction systems for driver assistance with experiments on language modeling for streets, hotels, and cities. Chapter 7 concentrates more on high speech error corrective coding for mobile phone applications which are of interest for car information access. Chapter 8 considers a speech enhancement method for noise suppression in the car environment. Chapter 9 seeks to integrate prosodic structure into noisy speech recognition applications. Effective noise reduction strategies for mobile and vehicle applications are considered in Chapter 10, and also in Chapter 11. Chapter 12 considers a small vocabulary speech system for controlling car environments. Chapters 13 and 14 consider transmission and compression schemes respectively for image and video applications which will become more critical for wireless information access within car environments in the near future. Chapter 15 follows up with a work on adaptive techniques for wireless speech transmission in local area networks, an area which will be critical if vehicles are to share information regarding road and weather conditions while on the road. Chapter 16 considers the use of audio-video information processing to help identify a speaker. This will have useful applications for driver identification in high noise conditions for the car. Chapter 17 considers a rather interesting idea of characterizing driving behavior based on biometric information including gas and brake pedal usage in the car. Chapter 18 addresses convolutional noise using blind signal separation for in-car environments. Finally, Chapter 19 develops a novel approach using multiple regression of the log spectra to model the differences between a close talking microphone and far-field microphone for in-vehicle applications.

Collectively, the research advances presented in these chapters offers a unique perspective of the state of the art for in-vehicle systems. The treatment of corpora, dialog system development, environmental noise suppression, hands-free microphone and array processing, integration of audio-video technologies, and wireless communications all point to the rapidly advancing field. From these studies, and others in the field from laboratories who were not able to participate in the DSP in Mobile and Vehicular Systems Workshop [http://dspincars.sdsu.edu/] in April 2003, it is clear that the domain of in-vehicle speech systems and information access is a rapidly advancing field with significant opportunities for advancement.

In closing, we would like to acknowledge the generous support from CIAIR for the DSP in Mobile and Vehicular Systems Workshop, and especially Professor Fumitada Itakura, who's vision and collaborative style in the field of speech processing has served as an example of how to bring together leading researchers in the field to share their ideas and work together for solutions to solve problems for in-vehicle speech and information systems.

Hüseyin Abut,
San Diego State Univ.

John H.L. Hansen,
Univ. Colorado at Boulder

Kazuya Takeda,
Nagoya University

This page intentionally left blank

Chapter 1

CONSTRUCTION AND ANALYSIS OF A MULTI-LAYERED IN-CAR SPOKEN DIALOGUE CORPUS

Nobuo Kawaguchi, Shigeki Matsubara, Itsuki Kishida, Yuki Irie, Hiroya Murao, Yukiko Yamaguchi, Kazuya Takeda and Fumitada Itakura
Center for Integrated Acoustic Information Research, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, JAPAN *Email: kawaguti@itc.nagoya-u.ac.jp*

Abstract: In this chapter, we will discuss the construction of the multi-layered in-car spoken dialogue corpus and the preliminary result of the analysis. We have developed the system specially built in a Data Collection Vehicle (DCV) which supports synchronous recording of multi-channel audio data from 16 microphones that can be placed in flexible positions, multi-channel video data from 3 cameras and the vehicle related data. Multimedia data has been collected for three sessions of spoken dialogue with different types of navigator in about 60-minute drive by each of 800 subjects. We have defined the Layered Intention Tag for the analysis of dialogue structure for each of speech unit. Then we have marked the tag to all of the dialogues for over 35,000 speech units. By using the dialogue sequence viewer we have developed, we can analyze the basic dialogue strategy of the human-navigator. We also report the preliminary analysis of the relation between the intention and linguistic phenomenon.

Keywords: Speech database, spoken dialogue corpus, intension tag, in-vehicle

1. INTRODUCTION

Spoken dialog interface using spontaneous speech is one of the most critical modules needed for effective hands-free human-machine interaction in vehicles when the safety is in mind. To develop framework for this, large-scale speech corpora play important roles for both of acoustic modelling and speech modelling in the field of robust and natural speech interface.

The Center for Integrated Acoustic Information Research (CIAIR) at Nagoya University has been developing a significantly large scale corpus for in-car speech applications [1,5,6]. Departing from earlier studies on the subject, the dynamic behaviour of the driver and the vehicle has been taken into account as well as the content of the in-car speech. These include the vehicle-specific data, driver-specific behavioural signals, the traffic conditions, and the distance to the destination [2,8,9]. In this chapter, details of this multimedia data collection effort will be presented. The main objectives of this data collection are as follows:

- Training acoustic models for the in-car speech data,
- Training language models of spoken dialogue for task domains related to information access while driving a car, and
- Modelling the communication by analyzing the interaction among different types of multimedia data.

In our project, a system specially developed in a Data Collection Vehicle (DCV) (Figure 1-1) has been used for synchronous recording of multi-channel audio signals, multi-channel video data, and the vehicle related information. Approximately, a total of 1.8 Terabytes of data has been collected by recording several sessions of spoken dialogue for about a period of 60-minutes drive by each of over 800 drivers. The driver gender breakdown is equal between the male and female drivers.

All of the spoken dialogues for each trip are transcribed with detailed information including a synchronized time stamp. We have introduced and employed a Layered Intention Tag (LIT) for analyzing dialogue structure. Hence, the data can be used for analyzing and modelling the interactions between the navigators and drivers involved in an in-car environment both under driving and idling conditions.

This chapter is organized as follows. In the next section, we describe the multimedia data collection procedure performed using our Data Collection Vehicle (DCV). In Section 3, we introduce the Layered Intention Tag (LIT) for analysis of dialogue scenarios. Section 4 briefly describes other layers of the corpus. Our preliminary findings are presented in Section 5.

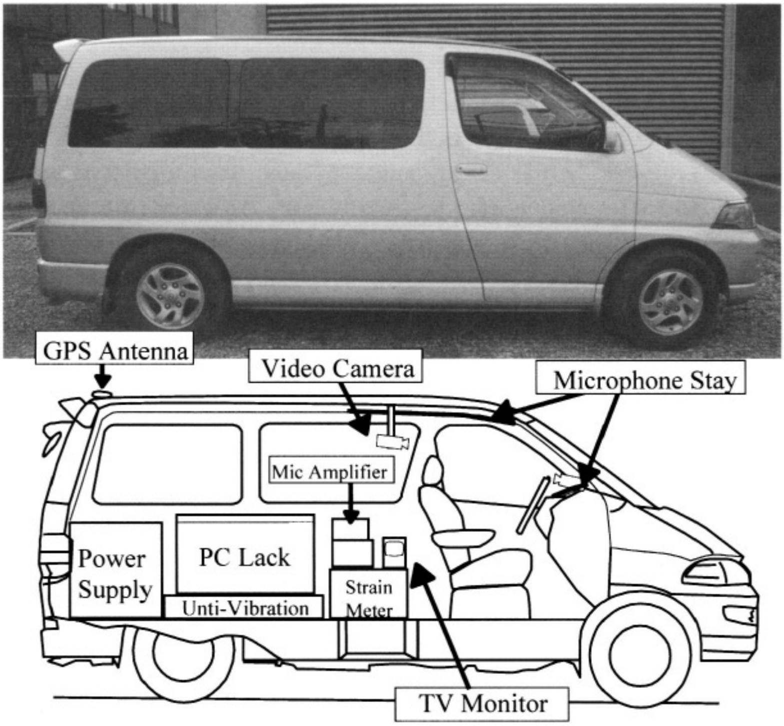


Figure 1-1. Data Collection Vehicle

2. IN-CAR SPEECH DATA COLLECTION

We have carried out our extensive data collection starting 1999 through 2001 over 800 subjects both under driving and idling conditions. The collected data types are shown in Table 1-1. In particular, during the first year, we have collected the following data from 212 subjects: (1) pseudo information retrieval dialogue between a subject and the human navigator, (2) phonetically balanced sentences, (3) isolated words, and (4) digit strings.

In the 2000-2001 collection, however, we have included two more dialogue modes such that each subject has completed a dialogue with three different kinds of interface systems. The first system is a human navigator, who sits in a special chamber inside the vehicle and the navigator converses and naturally. Second one is a wizard of Oz (WOZ) type system. The final one is an automatic dialog set-up based on automatic speech recognition (ASR). As it is normally done in many Japanese projects, we have employed Julius [3] as the ASR engine. In Table 1-2 we tabulate the driver age distribution.

Each subject has read 50 phonetically balanced sentences in the car while the vehicle was idling and subsequently drivers have spoken 25 sentences while driving the car. While idling, subjects have used a printed text posted on the dashboard to read a set of phonetically balanced sentences. While driving, we have employed a slightly different procedure for safety reasons. In this case, subjects are prompted for each phonetically sentence from a head-set utilizing a specially developed waveform playback software.

1999's collection	212.subj.
Spoken dialog with human navigator	11 min
PB sent. (Idling)	50 sent.
PB sent. (Driving)	25 sent.
Isolated words	30 words
Digit Strings	4digit*20
2000-2001's collection	300*2 subj.
Spoken dialog with human navigator	5min
Spoken dialog with WOZ system	5min
Spoken dialog with ASR system	5min
PB sent. (Idling)	50 sent.
PB sent. (Driving)	25 sent.
Isolated words	30 words
Digit Strings	4digit*20

Table 1-1. **Speech Data Specifications**

The recording system in our data collection vehicle is custom-designed equipment developed at CIAIR for this task. It is capable of synchronous recording of 12-channel audio inputs, 3-channel video data, and various vehicle related data. The recording system consists of eight network-connected computers, a number of distributed microphones and microphone amplifiers, a video monitor, three video cameras, a few pressure sensors, a differential-GPS unit, and an uninterruptible power supply (UPS).

Age	Male	Female	Sum
10--19	4	0	4
20--29	366	162	528
30--39	105	85	190
40--49	46	35	81
50--59	5	2	7
60--	2	0	2
Sum	528	284	812

Table 1-2. Driver age distribution.

Individual computers are used for speech input, sound output, three video channels, and vehicle related data. In Table 1-3, we list the recording characteristics of 12-speech and 3-video channels, five analog control signals from the vehicle representing the driving behavior of drivers and the location information from the DGPS unit built into the DCV. These multi-dimensional data are recorded synchronously, and hence, they can be synchronously analyzed.

Speech	16kHz, 16bit, 12ch
Video	MPEG-1, 29.97fps, 3ch
Control Signal	Status of Accelerator and Brake, Steering Wheel Angle Engine RPM, Speed: 16bit 1kHz
Location	Differential GPS (one reading per sec.)

Table 1-3. Recorded data specifications.

2.1 Multi-mode Dialogue Data Collection

The primary objective of the dialogue speech collection is to record three different modes of dialogue mentioned earlier. It is important to note that the task domain is the information retrieval task for all three modes. The descriptions of these dialogue modes are:

- **Dialogue with human navigator (HUM):** Navigators are trained in advance and has extensive information for the tasks involved. However, in order to avoid a dialogue divergence, some restriction is put on the way he/she speaks.

- **Dialogue with Wizard of OZ system (WOZ):** The WOZ mode is a spoken dialogue platform which has a touch-panel input for the human navigator and a speech synthesizer output. The system has a considerable list of shops and restaurants along the route and the navigator use the system to search and select the most suitable answer for subjects' spoken requests (Figure 1-2).

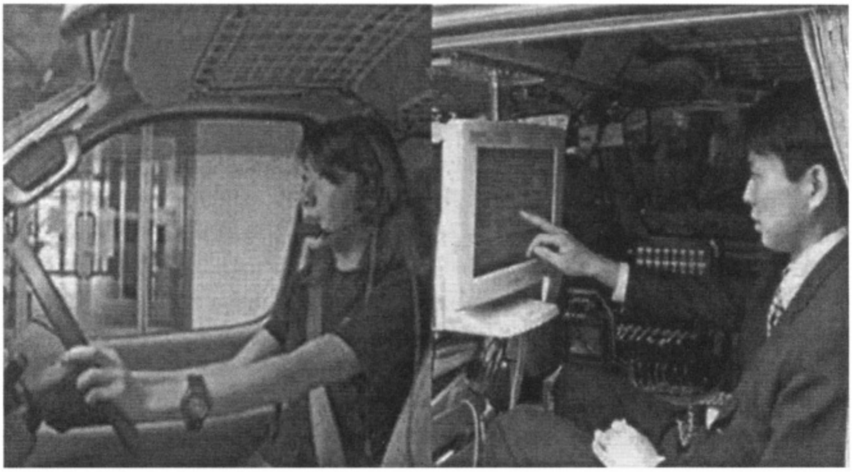


Figure 1-2. Sample Dialogue Recording Scene Using WOZ

- **Dialogue with Spoken Dialogue System (SYS):** The dialogue system called "Logger" performs a slot-filling dialogue for the restaurant retrieval task. The system utilizes Julius[3] for LVCSR system.

To simplify dialogue recording process, the navigator has prompted each task by using several levels of a task description panel to initiate the spontaneous speech. There is a number of task description panels associated with our task domain. A sample set from the task description panels are as follows:

'Fast food',
'Hungry',
'Hot summer, thirsty',
'No money', and
'You just returned from abroad'.

All of our recorded dialogues are transcribed into text in compliance with a set of criteria established for the Corpus of Spontaneous Japanese (CSJ) [13]. In Table 1-4, we tabulate many statistical data associated with our dialogue corpus. As it can be observed from the first row, we have collected more than 187 hours of speech data corresponding to approximately one million morpheme dialogue units.

2.2 Task Domains

We have categorized the sessions into several task domains. In Figure 1-3, we show the breakdown of major task domains. It is easy to see that approximately forty percent of the tasks are related to restaurant information retrieval, which is consistent with earlier studies. In the sections to follow, we will use only the data from the restaurant task. Our findings for other tasks and driver behavioral data will be discussed later Chapters 17 and 19.

	99HUM	00HUM	00WOZ	00SYS	01HUM	01WOZ	01SYS	Sum/ Ave.	Total Hours
Recording time(s)	141822	94692	95300	77922	93465	93862	78169	675232	187.6
Sessions	209	294	293	288	295	294	287	1960	
Average duration(s)	679	322	325	271	317	319	272		
Speech duration(s)	98100	69390	50864	54056	67635	47424	48877	436346	121.2
driver	44722	28085	20159	11515	26055	18127	11001	159664	44.4
operator	53328	41305	30705	42541	41580	29297	37876	276632	76.8
Speech unit	38760	25251	19585	24944	24178	19993	22904	175615	
driver	20493	12555	9381	10567	11985	9245	10722	84948	
operator	18267	12696	9754	14377	12193	10748	12182	90217	
Morphemes	252289	174848	107010	142674	176915	88459	124018	1066213	
driver	109710	68548	49023	27119	64173	44370	25587	388530	
operator	142579	106300	57987	115555	112742	44089	98431	677683	
Average morph/unit	6.71	7.18	5.64	5.93	7.70	4.60	5.65	6.31	
driver	5.65	5.78	5.31	2.79	5.79	5.24	2.61	4.88	
operator	7.85	8.52	5.97	8.07	9.48	4.10	8.10	7.58	
Ave. mora/sec	6.41	6.53	6.12	6.11	6.52	6.01	5.95	6.28	
driver	6.01	6.01	6.07	5.53	6.01	6.06	5.65	5.97	
operator	6.86	7.05	6.18	6.54	7.02	5.96	6.21	6.60	

Table 1-4. Corpus Statistics

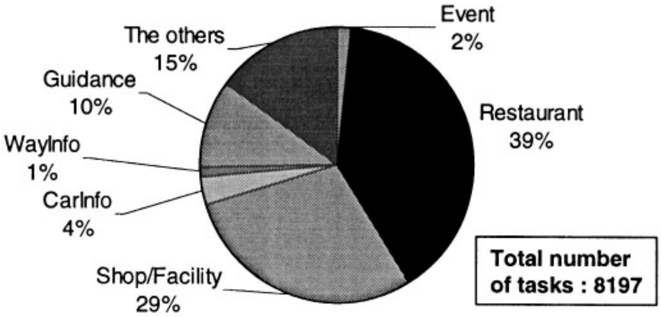


Figure 1-3. Task distribution of the corpus.

3. LAYERED INTENTION TAG

To develop a spoken dialogue system based on speech corpus [4], certain pre-specified information is required for each sentence corresponding to a particular response of the system. Additionally, to perform the response to satisfy the user, we need to presume the intention of the user's utterances. From our preliminary trials, we have learned that user's intention has a wide range even for a rather simple task, which could necessitate the creation of dozens of intention tags. To organize and expedite the process, we have stratified tags into several layers, which have resulted in an additional benefit of a hierarchical approach in analyzing users' intentions.

Our Layered Intention Tags (LIT) are described in Table 1-5 and the structure is shown in Figure 1-4. Each LIT is composed of four layers. The discourse act layer signifies the role of the speech unit in a given dialogue, which are labeled as "task independent tags". However, some units do not have a tag at this layer.

Action layer denotes the action taken. Action tags are subdivided into "task independent tags" and "task dependent tags". "Confirm" and "Exhibit" are task independent, whereas "Search", "ReSearch", "Guide", "Select" and "Reserve" are the task dependent ones.

Object layer stands for the objective of a given action including "Shop", "Parking."

Finally, the argument layer denotes other miscellaneous information about the speech unit. Argument layer is often decided directly from some specific keywords in a given sentence. As it is shown in Figure 1-4, the lower layered intention tags are explicitly depended on the upper layered ones.

Discourse Act	Action	Object	Argument
Request(Req)	Confirm(Conf)	Shop	ShopName
Propose(Prop)	Exhibit(Exhb)	Parking	Genre
Express(Expr)	Search(Srch)	ShopInfo	Price
Suggest(Sugg)	ReSearch(ReSe)	ParkingInfo	Place
Statement(Stat)	Guide(Guid)	SearchResult	Date
	Select(Sel)	RequestDetail	Menu
	Reserve(Res)	SelectionDetail	Count
		YesOrNo	Time

Table 1-5. Layered Intention Tag List (Partial)

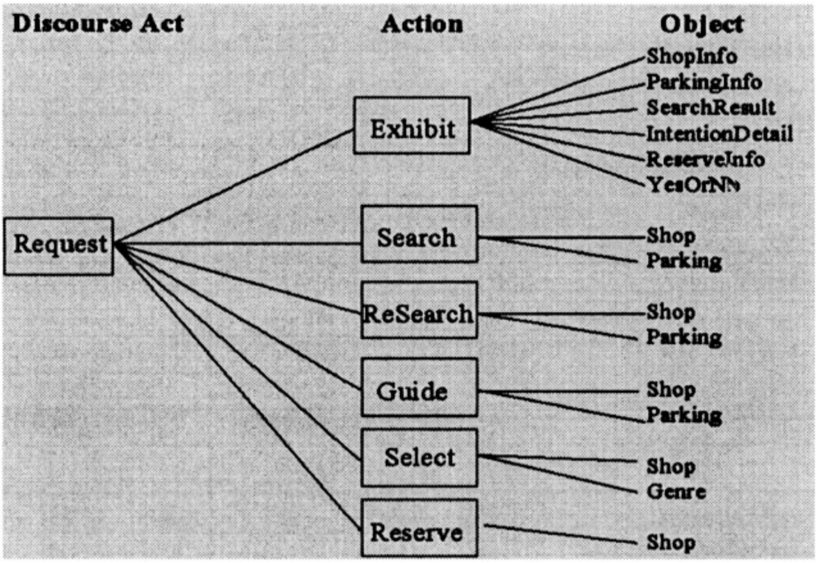
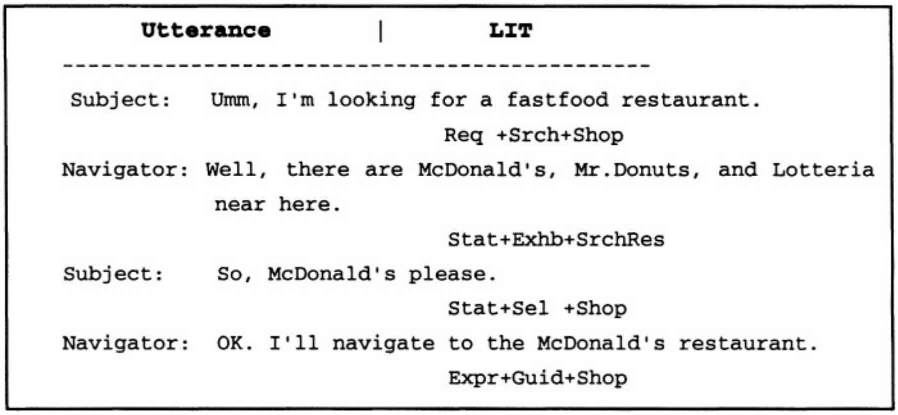


Figure 1-4. Structure of the Layered Intention Tag. (Partial List)

An example of a dialogue between a human navigator and a subject is shown in Figure 1-5. We have collected over 35,000 utterances (speech units) to be individually tagging in our restaurant information retrieval task. In Table 1-6, we present the statistics of the intention tagged corpus, resulting in a 3641 tagged tasks. It is thirty eight percent of the overall corpus. Top ten types of layered intention tags and their frequency of occurrence are given in Table 1-7. It is interesting to note that the tendencies of the tags are very similar in the recordings of both human navigator and the WOZ.



```
        Utterance          |         LIT
 ----------------------------------------------
 Subject:   Umm, I'm looking for a fastfood restaurant.
                                 Req +Srch+Shop
Navigator: Well, there are McDonald's, Mr.Donuts, and Lotteria
            near here.
                                 Stat+Exhb+SrchRes
Subject:    So, McDonald's please.
                                 Stat+Sel +Shop
Navigator:  OK. I'll navigate to the McDonald's restaurant.
                                 Expr+Guid+Shop
```

Figure 1-5. A sample dialogue transcription with its Layered Intention Tag.

	99HUM	00HUM	00WOZ	01HUM	01WOZ
Sessions	72	297	297	295	295
Task: Restaurant	425	793	890	626	907
Speech unit	4,909	8,133	8,420	5,628	8,331
driver	2,331	3,806	3,760	2,624	3,713
operator	2,578	4,327	4,660	3,004	4,618

Table 1-6. Statistics of the intention tagged corpus

	99	00	00	01	01	
	HUM	HUM	WOZ	HUM	WOZ	Total
Stat+Exhb+IntDetail	694	1,192	1,442	818	1,549	5,695
Stat+Exhb+SearchResult	665	1,303	1,260	938	1,285	5,451
Req+Srch+Shop	497	811	845	894	910	3,957
Expr+Guid+Shop	353	709	830	568	834	3,294
Stat+Sel+Shop	365	685	749	563	793	3,155
Stat+Exhb+ShopInfo	733	540	362	336	337	2,308
Req+Exhb+ShopInfo	655	377	223	259	338	1,852
Stat+Sel+Gerne	46	378	425	325	466	1,640
Req+Sel+Gerne	58	219	379	283	428	1,367
Req+ReSe+Shop	162	345	205	260	310	1,282

Table 1-7. Frequency of Occurrence of top ten intention tags

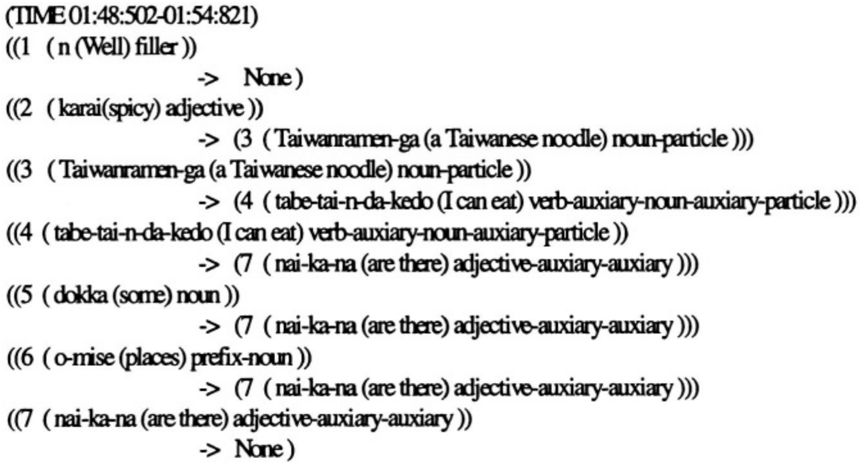
4. MULTI-LAYERED CORPUS STRUCTURE

Generally, a spoken dialogue system is developed by a sequence of signal processing sub-units. Starting with front end processing, such as filtering to avoid aliasing, acoustic-level signal processing including, sampling, quantization, and parameter extraction for ASR engine. Next the results from ASR are passed to a language processing unit based on a statistical language model appropriate for the tasks. Finally, there is an application-specific dialogue processing stage to carry out the tasks in a particular task domain such as the restaurant information retrieval application.

In order to use the collected dialogue data effectively in generating a comprehensive and robust corpus and then to update the system, not only a simple recording and transcription of speech are needed but a number of more advanced information is critically important. This necessitated undertaking a number of linguistic analyses on syntax and semantics of the corpus text. Thereby, a multi-layered spoken dialogue corpus of Figure 1-6 presented in the next section has become a method of choice to realize these.

4.1 Corpus with Dependency Tags

We have performed a dependency analysis to the drivers' utterances. Dependency in Japanese is a dependency relationship between the head of one bunsetsu and another bunsetsu. In addition, the bunsetsu, roughly corresponding to a basic phrase in English, is about the relation between an utterance intention and the utterance length, and the relation between utterance intentions and the underlying linguistic phenomena. Especially, we needed to pay attention to the smallest unit which a sentence can be divided naturally in terms of its meaning and pronunciation. It is also common to observe dependencies over two utterance units which are segmented by a pause. In this case, the dependency as a bunsetsu depending on a forward bunsetsu is the accepted norm. With these, we have carried out the data specification for spontaneous utterances. A sample of a corpus with dependency tags is shown in Figure 1-5. This corpus includes not only the dependency between bunsetsus but also the underlying morphological information, utterance unit information, dialogue turn information, and others. This corpus is used for acquisition of the dependency distribution for stochastic dependency parsing [14].



```
(TIME 01:48:502-01:54:821)
((1  ( n (Well) filler ))
                         ->   None )
((2  ( karai(spicy) adjective ))
                         ->  (3  ( Taiwanramen-ga (a Taiwanese noodle) noun-particle )))
((3  ( Taiwanramen-ga (a Taiwanese noodle) noun-particle ))
                         ->  (4  ( tabe-tai-n-da-kedo (I can eat) verb-auxiary-noun-auxiary-particle )))
((4  ( tabe-tai-n-da-kedo (I can eat) verb-auxiary-noun-auxiary-particle ))
                         ->  (7  ( nai-ka-na (are there) adjective-auxiary-auxiary )))
((5  ( dokka (some) noun ))
                         ->  (7  ( nai-ka-na (are there) adjective-auxiary-auxiary )))
((6  ( o-mise (places) prefix-noun ))
                         ->  (7  ( nai-ka-na (are there) adjective-auxiary-auxiary )))
((7  ( nai-ka-na (are there) adjective-auxiary-auxiary ))
                         ->  None )
```

Figure 1-6. Sample of the corpus with dependency tags.

5. DISCUSSION ON IN-CAR SPEECH CORPUS

In this section, we will be studying the characteristics of our multi-layered in-car speech corpus of Figure 1-7. In particular, we will explore the relationship between an intention in a given utterance and the utterance length, and the relationship between the intentions and the associated linguistic phenomenon. Especially, we will be comparing the driver's conversations with another person (human navigator) and the human-WOZ dialogues.

5.1 Dialogue Sequence Viewer

To understand and analyze the dialogue corpus intuitively, we have constructed a dialogue sequence viewer as depicted in Figure 1-8. For this task, we have also formed "turns" from speech units or tagged units, to indicate the event of a speaker change. In this figure, each node has a tag with a turn number, and the link between any two nodes implies the sequence of the event in a given conversation. As expected, each turn could have more than one LIT tag. The thickness of a link is associated with the occurrence count of a given tag's connections. For instance, there are only four turns in Figure 1-8 of the dialogue segment of Figure 1-4. We have observed that the average turn count in the restaurant query task is approximately 10.

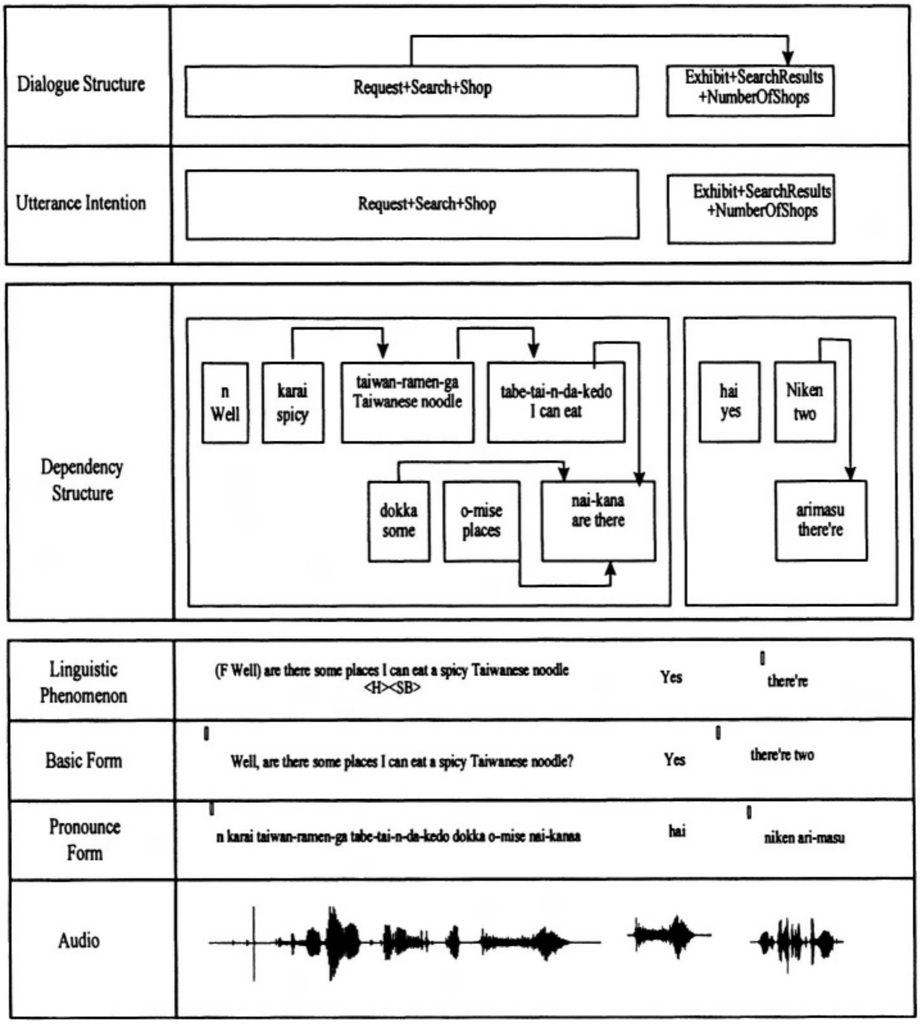

Dialogue Structure
Request+Search+Shop
Exhibit+SearchResults +NumberOfShops
Utterance Intention
Request+Search+Shop
Exhibit+SearchResults +NumberOfShops
Dependency Structure
n Well
karai spicy
taiwan-ramen-ga Taiwanese noodle
tabe-tai-n-da-kedo I can eat
dokka some
o-mise places
nai-kana are there
hai yes
Niken two
arimasu there're
Linguistic Phenomenon
(F Well) are there some places I can eat a spicy Taiwanese noodle <H><SB>
Yes
there're
Basic Form
Well, are there some places I can eat a spicy Taiwanese noodle?
Yes
there're two
Pronounce Form
n karai taiwan-ramen-ga tabe-tai-n-da-kedo dokka o-mise nai-kanaa
hai
niken ari-masu
Audio

Figure 1-7. Multi-Layered In-Car Speech Corpus Framework.

We found by employing the dialogue sequence viewer that the majority of the dialogue sequences pass through typical tags such as “Req+Srch+Shop”, “Stat+Exhb+SrchRes”, “Stat+Sel+Shop”, and “Expr+Guid+Shop.” We have also studied the dialogue segments with length 6, 8 and 10 turns. It turns out that the start section and the end section of dialogues of different lengths are very similar.

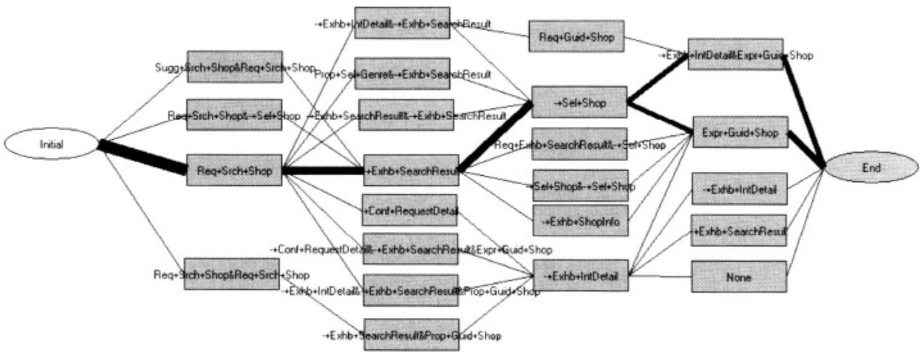

Initial
Sugg+Srch+Shop&Req+Srch+Shop
Req+Srch+Shop&-+Sel+Shop
Req+Srch+Shop
Req+Srch+Shop&Req+Srch+Shop
-+Exhb+IntDetail&-+Exhb+SearchResult
Prop+Sel+Genre&-+Exhb+SearchResult
-+Exhb+SearchResult&-+Exhb+SearchResult
+Exhb+SearchResult
+Conf+RequestDetail
-+Conf+RequestDetail&-+Exhb+SearchResult&Expr+Guid+Shop
-+Exhb+IntDetail&-+Exhb+SearchResult&Prop+Guid+Shop
-+Exhb+SearchResult&Prop+Guid+Shop
Req+Guid+Shop
-+Sel+Shop
Req+Exhb+SearchResult&-+Sel+Shop
-+Sel+Shop&-+Sel+Shop
-+Exhb+ShopInfo
-+Exhb+IntDetail
-+Exhb+IntDetail&Expr+Guid+Shop
Expr+Guid+Shop
-+Exhb+IntDetail
+Exhb+SearchResult
None
End

Figure 1-8. Partial dialogue sequence map in the Layered Intention Tag (LIT) Structure.

5.2 System Performance Differences between Human and WOZ Navigators

As discussed earlier, we have collected in-car information conversations using an actual human navigator, Wizard of Oz (WOZ) setup, and speech recognition (ASR) system. Since the ASR systems are configured to work in the system initiated conversation mode and they require considerable training to learn the speaking habits of drivers -more than 800 in our case- they were thought to be highly restricted this application. On the other hand, the human navigator and the WOZ setups have been observed to be very effective and easy to realize. Hence, we will be presenting some comparative results on driver behaviour if they converse with a human navigator or the Wizard of OZ system.

In Figure 1-9, we have plotted the performance of top ten layered intention tags (LIT) in the restaurant query task, where lines represent the number of phrases per speech unit in the case of human navigator (H) and the WOZ system (W). We have also included a bar diagram for the occurrence rate of linguistic fillers.

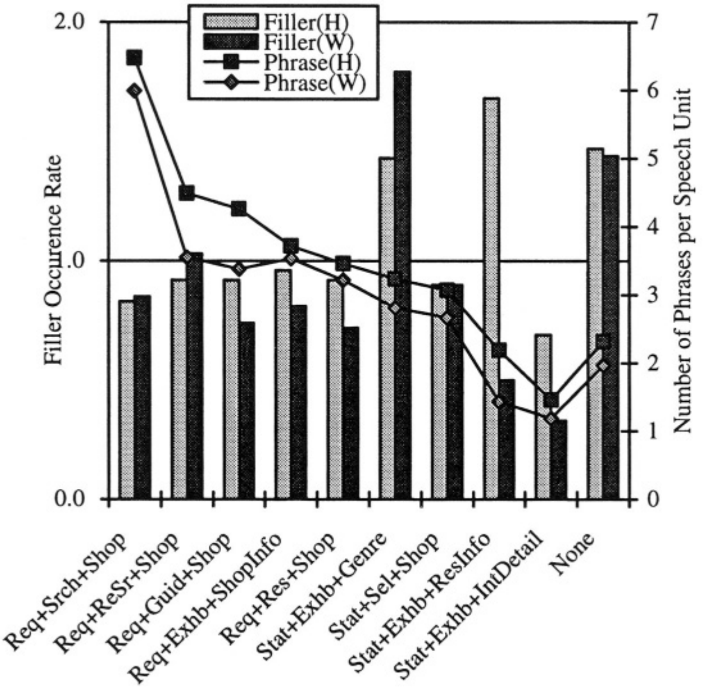


Figure 1-9. Driver Behaviour Differences between HUM and WOZ navigators.

Average occurrence of filler was 0.15 per phrase in the case of a human navigator and the corresponding rate was 0.12 per phrase in the WOZ case. Therefore, we can conclude that the average dialogue between a driver and the WOZ is shorter than that of a human navigator. This tendency is observed to be fairly uniform across all the LITs under study.

The occurrence rate of filler for "Request(Req)" tags is close to average. Although other tags show sizeable differences, there was not any difference between the human navigator and then WOZ setup. The differences were consistently high in other tags. This means that, for the "Req" tags, subjects frequently tend to speak regardless of the reply from the system. On the other hand, subjects simply tend to respond to an utterance from the system for other tags. It is fairly probable that the fluency of the system could affect the driver's speech significantly.

Finally, we could also conclude from the number of phrases per speech unit that the "Req" tagged units are highly complex sentences in comparison to other tagged units.

6. SUMMARY

In this chapter, we have presented brief description of a multimedia corpus of in-car speech communication developed in CIAIR at Nagoya University, Japan. The corpus consists of synchronously recorded multi-channel audio/video signals, driving signals, and a differential GPS reading. For a restaurant information query task domain speech dialogues were collected from over 800 drivers -equal split between male and female drivers- in four different modes, namely, human-human and human-machine, prompted, and natural. In addition, we have experimented with an ASR system for collecting human-machine dialogues. Every spoken dialogue is transcribed with precise time stamp.

We have proposed the concept of a Layered Intention Tag (LIT) for sequential analysis of dialogue speech. Towards that end, we have tagged one half of the complete corpus with LITs. We have also attached structured dependency information to the corpus. With these, in-car speech dialogue corpus has been enriched to turn into a multi-layered corpus. By studying different layers of the corpus, different aspects of the dialogue can be analyzed.

Currently, we are exploring the relationship between an LIT and the number of phrases and the occurrence rate of fillers with an objective of developing a corpus based dialogue management platform.

ACKNOWLEDGEMENT

This work has been supported in part by a Grant-in-Aid for Center of Excellence (COE) Research No. 11CE2005 from the Ministry of Education, Science, Sports and Culture, Japan. The authors would like to acknowledge the members of CIAIR for their enormous contribution and efforts towards the construction of the in-car spoken dialogue corpus.

REFERENCES

[1] Nobuo Kawaguchi, Shigeki Matsubara, Kazuya Takeda, and Fumitada Itakura: Multimedia Data Collection of In-Car Speech Communication, Proc. of the 7th European Conference on Speech Communication and Technology (EUROSPEECH2001), pp. 2027--2030, Sep. 2001, Aalborg.

[2] Deb Roy: "Grounded" Speech Communication, Proc. of the International Conference on Spoken Language Processing (ICSLP 2000), pp.IV69--IV72, 2000, Beijing.

[3] T.Kawahara, T.Kobayashi, K.Takeda, N.Minematsu K.Itou, M.Yamamoto, A.Yamada, T.Utsuro, K.Shikano : Japanese Dictation Toolkit: Plug-and-play Framework For Speech Recognition R&D, Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'99), pp.393--396 (1999).

[4] Hiroya Murao, Nobuo Kawaguchi, Shigeki Matsubara, and Yasuyoshi Inagaki: Example-Based Query Generation for Spontaneous Speech, Proc. of the 7th IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU01), Dec.2001, Madonna di Campiglio.

[5] Nobuo Kawaguchi, Kazuya Takeda, Shigeki Matsubara, Ikuya Yokoo, Taisuke Ito, Kiyoshi Tatara, Tetsuya Shinde and Fumitada Itakura, CIAIR speech corpus for real world speech recognition, Proceedings of 5th Symposium on Natural Language Processing (SNLP-2002) & Oriental COCOSDA Workshop 2002, pp. 288-295, May 2002, Hua Hin, Thailand.

[6] Nobuo Kawaguchi, Shigeki Matsubara, Kazuya Takeda, and Fumitada Itakura, Multi-Dimensional Data Acquisition for Integrated Acoustic Information Research, Proceedings of 3rd International Conference on Language Resources and Evaluation (LREC-2002), Vol.: I, pp. 2043-2046, May 2002, Canary Islands.

[7] Shigeki Matsubara, Shinichi Kimura, Nobuo Kawaguchi, Yukiko Yamaguchi and Yasuyoshi Inagaki : Example-based Speech Intention Understanding and Its Application to In-Car Spoken Dialogue System, Proceedings of the 17th International Conference on Computational Linguistics (COLING-2002), Vol. 1, pp. 633-639, Aug. 2002, Taipei.

[8] J. Hansen, P. Angkititrakul, J. Plucienkowski, S.Gallant, U. Yapanel, B. Pellom, W. Ward, and R. Cole: "CU-Move": Analysis & Corpus Development for Interactive In-Vehicle Speech Systems, Proc. of the 7th European Conference on Speech Communication and Technology (EUROSPEECH2001), pp. 2023--2026, Sep. 2001, Aalborg.

[9] P. A. Heeman, D. Cole, and A. Cronk: The U.S. SpeechDat-Car Data Collection, Proceedings of the Seventh European Conference on Speech Communication and Technology (EUROSPEECH2001), pp. 2031--2034, Sep. 2001, Aalborg.

[10] CIAIR home page : http://www.ciair.coe.nagoya-u.ac.jp

[11] Yuki Irie, Nobuo Kawaguchi, Shigeki Matsubara, Itsuki Kishida, Yukiko Yamaguchi, Kazuya Takeda, Fumitada Itakura, and Yasuyoshi Inagaki: An Advanced Japanese Speech Corpus for In-Car Spoken Dialogue Research, in Proceedings of of Oriental COCOSDA-2003, pp.209—216(2003).

[12] Itsuki Kishida, Yuki Irie, Yukiko Yamaguchi, Shigeki Matsubara, Nobuo Kawaguchi and Yasuyoshi Inagaki: Construction of an Advanced In-Car Spoken Dialogue Corpus and its Characteristic Analysis, in Proc. of EUROSPEECH2003, pp.1581—1584(2003).

[13] K. Maekawa, H. Koiso, S. Furui, H. Isahara, "Spontaneous Speech Corpus of Japanese", in Proc. of LREC-2000, No.262(2000).

[14] S. Matsubara, T. Murase, N. Kawaguchi and Y. Inagaki,"Stochastic Dependency Parsing of Spontaneous Japanese Spoken Language", Proc. of COLING-2002, Vol.1, pp.640-645(2002).

This page intentionally left blank

Chapter 2

CU-MOVE: ADVANCED IN-VEHICLE SPEECH SYSTEMS FOR ROUTE NAVIGATION[1]

John H.L. Hansen, Xianxian Zhang, Murat Akbacak, Umit H. Yapanel, Bryan Pellom, Wayne Ward, Pongtep Angkititrakul
Robust Speech Processing Group, Center for Spoken Language Research, University of Colorado at Boulder, Boulder, Colorado 80309-0594, USA
Email: John.Hansen@colorado. edu

Abstract: In this chapter, we present our recent advances in the formulation and development of an in-vehicle hands-free route navigation system. The system is comprised of a multi-microphone array processing front-end, environmental sniffer (for noise analysis), robust speech recognition system, and dialog manager and information servers. We also present our recently completed speech corpus for in-vehicle interactive speech systems for route planning and navigation. The corpus consists of five domains which include: digit strings, route navigation expressions, street and location sentences, phonetically balanced sentences, and a route navigation dialog in a human Wizard-of-Oz like scenario. A total of 500 speakers were collected from across the United States of America during a six month period from April-Sept. 2001. While previous attempts at in-vehicle speech systems have generally focused on isolated command words to set radio frequencies, temperature control, etc., the CU-Move system is focused on natural conversational interaction between the user and in-vehicle system. After presenting our proposed in-vehicle speech system, we consider advances in multi-channel array processing, environmental noise

[1] This work was supported in part by DARPA through SPAWAR under Grant No. N66001-002-8906, from SPAWAR under Grant No. N66001-03-1-8905, in part by NSF under Cooperative Agreement No. IIS-9817485, and in part by CSLR Center Member support from Motorola, HRL, Toyota CR&D, and CSLR Corpus Member support from SpeechWorks, Infinitive Speech Systems (Visteon Corp.), Mitsubishi Electric Research Lab, Panasonic Speech Technology Lab, and VoiceSignal Technologies.

sniffing and tracking, new and more robust acoustic front-end representations and built-in speaker normalization for robust ASR, and our back-end dialog navigation information retrieval sub-system connected to the WWW. Results are presented in each sub-section with a discussion at the end of the chapter.

Keywords: Automatic speech recognition, robustness, microphone array processing, multi-modal, speech enhancement, environmental sniffing, PMVDR features, dialog, mobile, route navigation, in-vehicle

1. INTRODUCTION: HANDS-FREE SPEECH RECOGNITION/DIALOG IN CARS

There has been significant interest in the development of effective dialog systems in diverse environmental conditions. One application which has received much attention is for hands-free dialog systems in cars to allow the driver to stay focused on operating the vehicle while either speaking via cellular communications, command and control of vehicle functions (i.e., adjust radio, temperature controls, etc.), or accessing information via wireless connection (i.e., listening to voice mail, voice dialog for route navigation and planning). Today, many web based voice portals exist for managing call center and voice tasks. Also, a number of spoken document retrieval systems are available for information access to recent broadcast news content including SpeechBot by HP-Compaq)[30] and the SpeechFind for historical digital library audio content (RSPG-CSLR, Univ. Colorado)[29]. Access to audio content via wireless connections is desirable in both commercial vehicle environments (i.e., obtaining information on weather, driving conditions, business locations, etc.), points of interest and historical content (i.e., obtaining audio recordings which provide a narrative of historical places for vacations, etc.), as well as in military environments (i.e., information access for coordinating peacekeeping groups, etc.).

This chapter presents our recent activity in the formulation of a new in-vehicle interactive system for route planning and navigation. The system employs a number of speech processing sub-systems previously developed for the DARPA CU Communicator[1] (i.e., natural language parser, speech recognition, confidence measurement, text-to-speech synthesis, dialog manager, natural language generation, audio server). The proposed CU-Move system is an in-vehicle, naturally spoken mixed initiative dialog system to obtain real-time navigation and route planning information using GPS and information retrieval from the WWW. A proto-type in-vehicle platform was developed for speech corpora collection and system development. This includes the development of robust data collection and front-end processing for recognition model training and adaptation, as well as a back-end information server to obtain interactive automobile route planning information from WWW.

The novel aspects presented in this chapter include the formulation of a new microphone array and multi-channel noise suppression front-end, environmental (sniffer) classification for changing in-vehicle noise conditions, and a back-end navigation information retrieval task. We also discuss aspects of corpus development. Most multi-channel data acquisition algorithms focus merely on standard delay-and-sum beamforming methods. The new noise robust speech processing system uses a five-channel array with a constrained switched adaptive beamformer for the speech and a second for the noise. The speech adaptive beamformer and noise adaptive beamformer work together to suppress interference prior to the speech recognition task. The processing employed is capable of improving SegSNR performance by more than 10dB, and thereby suppress background noise sources inside the car environment (e.g., road noise from passing cars, wind noise from open windows, turn signals, air conditioning noise, etc.).

This chapter is organized as follows. In Sec. 2, we present our proposed in-vehicle system. In Sec. 3, we discuss the CU-Move corpus. In Sec. 4, we consider advances in array processing, followed by environmental sniffing, and automatic speech recognition (ASR), and our dialog system with connections to WWW. Sec. 5 concludes with a summary and discussion of areas for future work.

2. CU-MOVE SYSTEM FORMULATION

The problem of voice dialog within vehicle environments offers some important speech research challenges. Speech recognition in car environments is in general fragile, with word-error-rates (WER) ranging from 30-65% depending on driving conditions. These changing environmental conditions include speaker changes (task stress, emotion, Lombard effect, etc.)[16,31] as well as the acoustic environment (road/wind noise from windows, air conditioning, engine noise, exterior traffic, etc.).

Recent approaches to speech recognition in car environments have included combinations of basic HMM recognizers with front-end noise suppression[2,4], environmental noise adaptation, and multi-channel concepts. Many early approaches to speech recognition in the car focused on isolated commands. One study considered a command word scenario in car environments where an HMM was compared to a hidden Neural Network based recognizer[5]. Another method showed an improvement in computational requirements with front-end signal-subspace enhancement used a DCT in place of a KLT to better map speech features, with recognition rates increasing by 3-5% depending on driving conditions[6]. Another study[7] considered experiments to determine the impact of mismatch between recognizer training and testing using clean data, clean data with car noise added, and actual noisy car data. The results showed that starting with simulated noisy environment train models, about twice as much adaptation material is needed compared with starting with clean reference models. The work was later extended[8] to consider unsupervised online adaptation using previously formulated MLLR and MAP techniques. Endpoint detection of phrases for speech recognition in car environments has also been considered[9]. Preliminary speech/noise detection with front-end speech enhancement methods as noise suppression front-ends for robust speech recognition have also shown promise[2,4,10,11]. Recent work has also been devoted to speech data collection in car environments including SpeechDat.Car[12], and others [13]. These data concentrate primarily on isolated command words, city names, digits, etc. and typically do not include spontaneous speech for truly interactive dialogue systems. While speech recognition efforts in car environments generally focus on isolated word systems for command and control, there has been some work on developing more spontaneous speech based systems for car navigation [14,15], however these studies use a head-worn and ceiling mounted microphones for speech collection and limit the degree of naturalness (i.e., level of scripting) for navigation information exchange.

In developing CU-Move, there are a number of research challenges which must be addressed to achieve reliable and natural voice interaction within the car environment. Since the speaker is performing a task (driving the vehicle), a measured level of user task stress will be experienced by the driver and therefore this should be included in the speaker modeling phase. Previous studies have clearly shown that the effects of speaker stress and Lombard effect (i.e., speaking in noise) can cause speech recognition systems to fail rapidly[16]. In addition, microphone type and placement for in-vehicle speech collection can impact the level of acoustic background noise and ultimately speech recognition performance. Figure 2-1 shows a flow diagram of the proposed CU-Move system. The system consists of front-end speech collection/processing tasks that feed into the speech recognizer. The speech recognizer is an integral part of the dialogue system (tasks for Understanding, Discourse, Dialogue Management, Text Generation, and TTS). An image of the microphone used in the array construction is also shown (Figure 2-2). The back-end processing consists of the information server, route database, route planner, and interface with the navigation database and navigation guidance systems. Here, we focus on our efforts in multi-channel noise suppression, automatic environmental characterization, robust speech recognition, and a proto-type navigation dialogue.

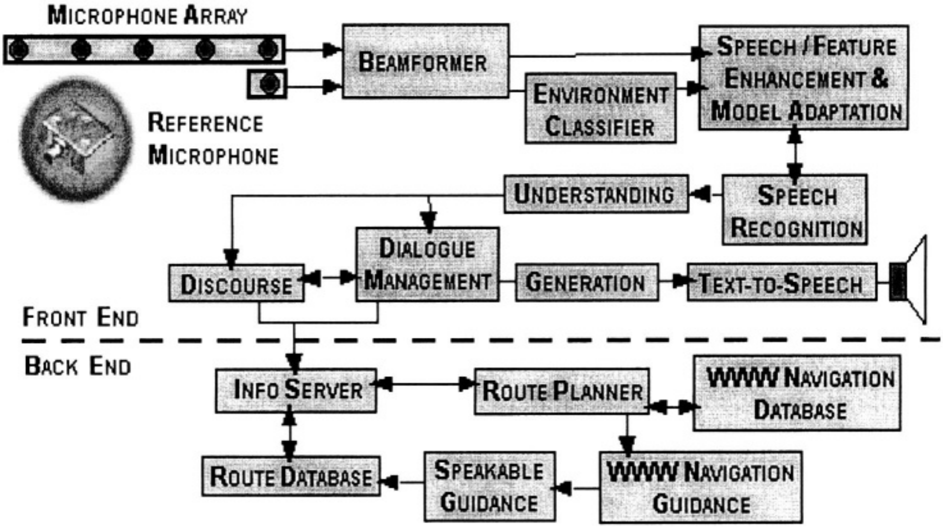


Figure 2-1. Flow Diagram of CU-Move Interactive Dialogue System for In-Vehicle Route Navigation

3. CU-MOVE CORPUS DEVELOPMENT

As part of the CU-Move system formulation, a two phase data collection plan was developed. Phase I focused on collecting acoustic noise and probe speech from a variety of cars and driving conditions. The outcome of Phase I was to determine the range of noise conditions across vehicles, and select one vehicle for Phase II collection that is representative of the typical noise domains experienced while driving. Eight vehicles were used in Phase I analysis (e.g., compact and two mid-size cars, small and medium pickup trucks, passenger van, sport utility vehicle (SUV), cargo van). We considered 14 noise conditions in actually driving scenarios. Figure 2-2 summarizes some of the results obtained from the study, with further details presented in [26]. The noise level was highest with windows open 2 inches traveling 65mph on the highway, and most quiet when the car was idle at a stop light. After detailed analysis, we determined the SUV represented the mid-range noise conditions (noise levels were high for compact cars and low for pickup trucks).

Next, Phase II speech collection was performed. Since the speaker is experiencing some level of stress by performing the task of driving the vehicle, this should be included in the speaker modeling phase. While Lombard effect can be employed, local state and federal laws in the United States limit the ability to allow subjects in this data collection to operate the vehicle and read prompts from a display. We therefore have subjects seated in the passenger seat, with prompts given on a small flat panel display attached to the dashboard to encourage subjects to stay focused on the roadway ahead. Speech data collection was performed across 6 U.S. cities that reflect regional dialects. These cities were selected to be mid-size cities, in order to increase the prospects of obtaining subjects who are native to that region. A balance across gender and age brackets was also maintained. The driver performed a fixed route similar to what was done for Phase I data collection so that a complete combination of driving conditions (city, highway, traffic noise, etc.) was included. The format of the data collection consists of five domains with four *Structured Text Prompt* sections and one *Wizard-of-Oz* (WOZ) dialog section:

Navigation Phrases: collection of phrases useful for In-Vehicle navigation interaction [prompts are fixed for all speakers]. Examples include: “Where is the closest gas station?” “How do I get to 1352 Pine Street?” “Which exit do I take?” “Is it a right or left turn?” “How do I get to the airport?” “I’m lost. Help me.”

Digit Sequences: each speaker produced 16 digit strings from a randomized 75 digital string set. Examples include: telephone numbers (425-952-54o0), random credit card numbers (1234-5621-1253-5981), and individual numbers (0,o,#86, *551).

Say and Spell Addresses: a randomized set of 20 strings of words/addresses were produced, with street names spelled. Some street names are used for all cities, some were drawn from local city maps. Examples include: Park Place, Ivy Circle, 3215 Marine Street, 902 Olympic Boulevard.

Phonetically Balanced Sentences: each speaker produced a collection of between 20-30 phonetically balanced from a set of 2500 sentences [prompts are randomized]. Examples include: “This was easy for us.” “Jane may earn more money by working hard.”

Dialog Wizard - of - Oz Collection: each speaker from the field called an on-line navigation system at CSLR, where a human wizard-of-oz like (WOZ) operator would guide the caller through three different navigation routes determined for that city. More than 100 potential destinations were previously established for each city between the driver and WOZ human operator, where detailed route information was stored for the operator to refer to while the caller was on the in-vehicle cell-phone. The list of establishments for that city were points of interest, restaurants, major intersections, etc. (e.g., “How do I get to the closest police station?”, “How do I get to the Hello Deli?”). The user calls using a modified cell-phone in the car, that allows for data collection using one of the digital channels from our recorder. The dialog was also recorded at CSLR where the WOZ operator was interacting with the field subject.

The 500 speaker corpus was fully transcribed, labeled, spell checked, beamformed/processed and organized for distribution. The un-processed version contains well over 600GB of data, and the processed version consists of a hard-disk release of approximately 200GB. Figure 2-3 shows the age distribution of the CU-Move corpus (further details presented in [26,27]).

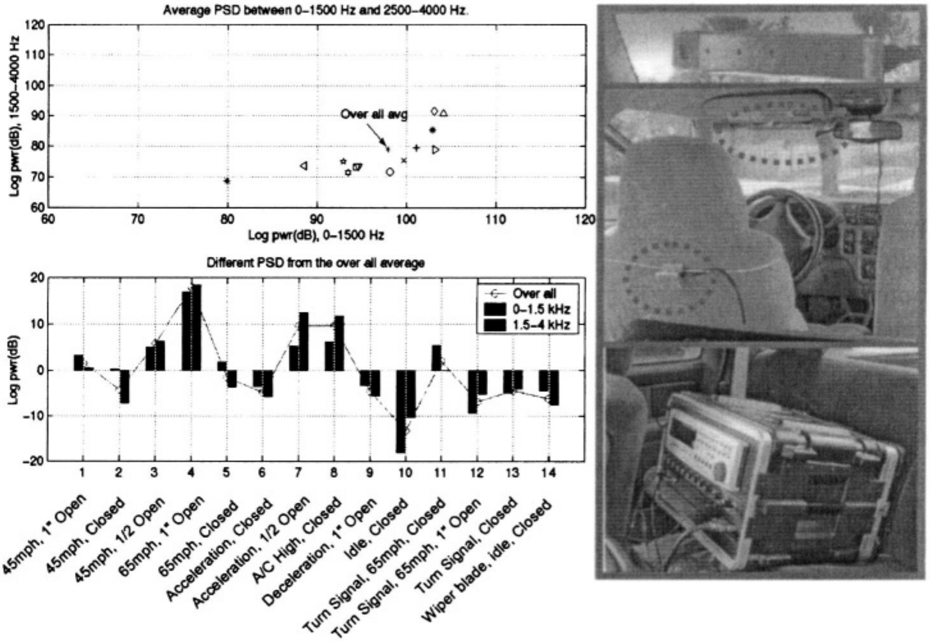


Figure 2-2. (a) Analysis of average power spectral density for low (0-1.5kHz) and high (1.5-4.0kHz) frequency bands for 14 car noise conditions from Phase-I data collection. Overall average noise level is also shown. (b) Photos show corpus collection setup: constructed microphone array (using Knowles microphones), array and reference microphone placement, constructed multi-channel DAT recorder (Fostex) with channel dependent level control and DC-to-AC converter.

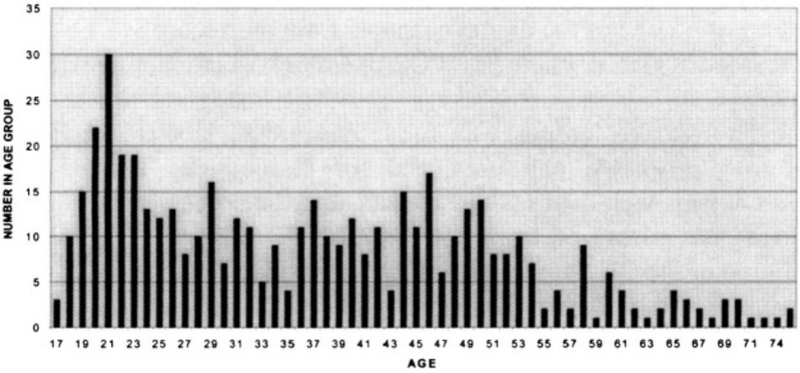


Figure 2-3. Age distribution (17-76 years old) of the 500 speaker CU-Move Corpus

In addition, a number of cross-transcriber reliability evaluations have been completed on the CU-Move corpus. Three transcribers were on the average, in agreement the majority of the time for parts 1-4 (prompts), with a 1.8% substitution rate when comparing transcriber hypotheses two at a time. When we consider the spontaneous route navigation WOZ part, transcriber files naturally had a higher difference, with a substitution rate of 3.0%. These numbers will depend on the clarity and articulation characteristics of the speakers across the six CU-Move dialect regions.

4. IN-VEHICLE SUB-SYSTEM FORMULATION

In this section, we discuss the formulation of our microphone array processing front-end, environmental sniffing, robust speech recognition system, and proto-type dialogue system.

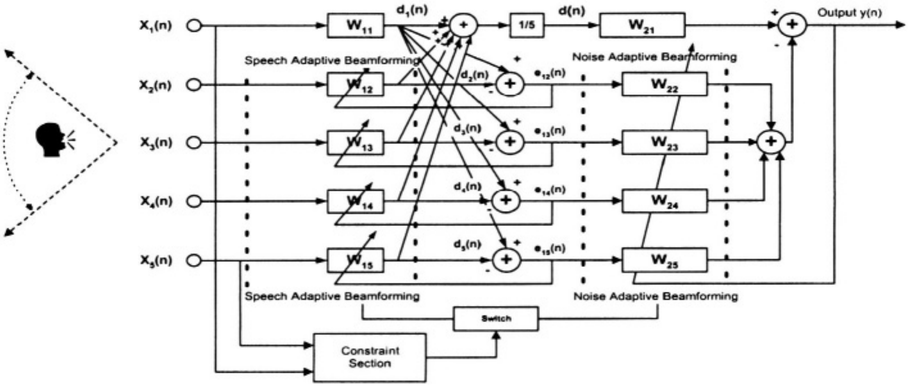


Figure 2-4. Flow diagram of the Proposed Constrained Switched Adaptive Beamforming (CSA -BF) algorithm.

4.1 Constrained Switched Adaptive Array-Processing (CSA-BF)

The proposed CSA-BF array processing algorithm consists of four parts: a constraint section (CS), a speech adaptive beamformer (SA-BF), a noise adaptive beamformer (NA-BF) and a switch. Figure 2-4 shows the detailed structure of CSA-BF, for a 5-microphone array. The CS is designed to identify potential speech and noise locations. If a speech source is detected, the switch will activate SA-BF to adjust the beam pattern and enhance the desired speech. At the same time, NA-BF is disabled to avoid speech leakage. If however, a noise source is detected, the switch will activate NA-BF to adjust the beam pattern for noise and switch off SA-BF processing to avoid the speech beam pattern from being altered by the noise. The combination of SA-BF and NA-BF processing results in a framework that achieves noise cancellation for interference in both time and spatial orientation. Next, we consider each processing stage of the proposed CSA-BF scheme.

4.1.1 Constraint Section

Many source localization methods have been considered in the past with effective performance for large microphone arrays in conference rooms or large auditoriums. Their ability to perform well in changing noisy car conditions has not been documented to the same degree, but is expected to be poor. Here, we propose three practical constraints that can be used to separate speech and noise sources with high accuracy.

- *Criterion 1 (Maximum averaged energy):* Since speech coming from the driver's direction will have on average the highest intensity of all sources present, therefore, we calculate the averaged signal TEO energy [18] frame by frame, and if this energy is greater than some threshold (Please refer to [19] for the details, we take the current signal frame as speech candidate.
- *Criterion 2 (LMS adaptive filter):* In order to separate the front-seat driver and passenger, we choose the adaptive LMS filter method and incorporate the geometric structure of the microphone array to locate the source.
- *Criterion 3 (Bump noise detector)* This final criterion is set to avoid instability in the filtering process which is affected by impulsive noise with high-energy content, such as road impulse/bump noise.

Finally, we note that the signal is labeled as speech if and only if all three criteria are satisfied.

4.1.2 Speech Adaptive Beamformer (SA-BF)

The function of SA-BF is to form an appropriate beam pattern to enhance the speech signal. Since adaptive filters are used to perform the beam steering, we can change beam pattern with a movement of the source. The degree of adaptation steering speed is decided by the convergence behavior of the adaptive filters. In our implementation, we select microphone 1 as the primary microphone, and build an adaptive filter between it and each of the other four microphones. These filters compensate for the different transfer functions between the speaker and the microphone array. A normalized LMS algorithm updates the filter coefficients only when the current signal is detected as speech. There are two kinds of output from the SA-BF: namely the enhanced speech $d(n)$ and noise signal $e_{1i}(n)$, which are given as follows,

$$d(n) = \frac{1}{5}\sum_{i=1}^{5} \mathbf{w}_{1i}^{T}(n)\mathbf{x}_{1i}(n) \tag{1}$$

$$e_{1i}(n) = \mathbf{w}_{11}^{T}(n)\mathbf{x}_{1}(n) - \mathbf{w}_{1i}^{T}(n)\mathbf{x}_{i}(n) \tag{2}$$

$$\mathbf{w}_{1i}(n+1) = \mathbf{w}_{1i}(n) + \frac{2\mu}{\mathbf{x}_{i}^{T}(n)\mathbf{x}_{i}(n)} e_{1i}(n)\mathbf{x}_{i}(n) \tag{3}$$

for channels i=2,3,4,5, where $\mathbf{w}_{11}(n)$ is a fixed filter.

4.1.3 Noise Adaptive Beamformer (NA-BF)

The NA-BF processor operates in a scheme like a multiple noise canceller, in which both the reference speech signal of the noise canceller and the speech free noise references are provided by the output of the SA-BF. Since the filter coefficients W_{2i} are updated only when the current signal is detected as noise, they form a beam that is directed towards the noise, thus the reason to name it a noise adaptive beamformer (NA-BF). The output response is given as,

$$y(n) = \mathbf{d}\ (n)\mathbf{w}_{21}^{T}(n) - \sum_{i=2}^{5} \mathbf{w}_{2i}^{T}(n)\mathbf{e}_{1i}(n) \tag{4}$$

$$\mathbf{w}_{2i}(n+1) = \mathbf{w}_{2i}(n) + \frac{2\mu}{\mathbf{e}_{1i}^{T}(n)\mathbf{e}_{1i}(n)} e_{1i}(n)d(n) \tag{5}$$

for microphone channels *i*=2,3,4,5.

4.1.4 Experimental Evaluation

In order to evaluate the performance of the CSA-BF algorithms in noisy car environments, we process all available speakers in Release 1.1a [21,26,27] of the CU-Move corpus using both CSA-BF and DASB algorithms, and compared the results. This release consists of 153 speakers, of which 117 were from the Minneapolis, MN area. We selected 67 of these speakers that include 28 males and 39 females, which reflects 8 hours of data. In order to compare the result of CSA-BF with that of DASB thoroughly, we also investigated the enhanced speech output from SA-BF. For evaluation, we consider two different performance measures using CU-Move data. One measure is the Segmental Signal-to-Noise Ratio (SegSNR) [22] which represents a noise reduction criterion for voice communications. The second performance measure is Word Error Rate (WER) reduction, which reflects benefits for speech recognition applications. The Sonic Recognizer [23,25] is used to investigate speech recognition performance. During the recognizer evaluation, we used 49 speakers (23 male, 26 female) as the training set, and 18 speakers (13 male, 5 female) as the test set.

Table 2-1 summarizes average SegSNR improvement, average WER, CORR (word correct rate), SUB (Word Substitution Rate), DEL (Word Deletion Rate) and INS (Word Insertion Rate). Here, the task was on the digits portion of CU-Move corpus (further details are presented in [19]). Figure 2-5 illustrates average SegSNR improvement and WER speech recognition performance results. The average SegSNR results are indicated by the bars using the left-side vertical scale (dB), and the WER improvement is the solid line using the right-side scale (%).

Method / Measure	chan3	DASB	SA-BF	CSA-BF
Ave. (dB) SegSNR	9.35	10.24	10.51	14.79
WER (%)	14.8	11.9	12	11
SUB (%)	7.9	6.8	6.6	6.2
DEL (%)	4.3	2.5	2.8	2.5
INS (%)	2.5	2.6	2.5	2.4
CORR (%)	87.7	90.7	90.5	91.3

Table 2-1. Average SegSNR (segmental signal-to-noise ratio), WER (word-error-rate), CORR (word correct rate), SUB (word substitution rate), DEL (word deletion rate) and INS (word insertion rate) for Reference Channel 3 Microphone (chan3) and three Array/Beamforming Scenarios: DASB (delay-and-sum beamforming), SA-BF (speech adaptive beamforming), CSA-BF (constrained switched adaptive beamforming).

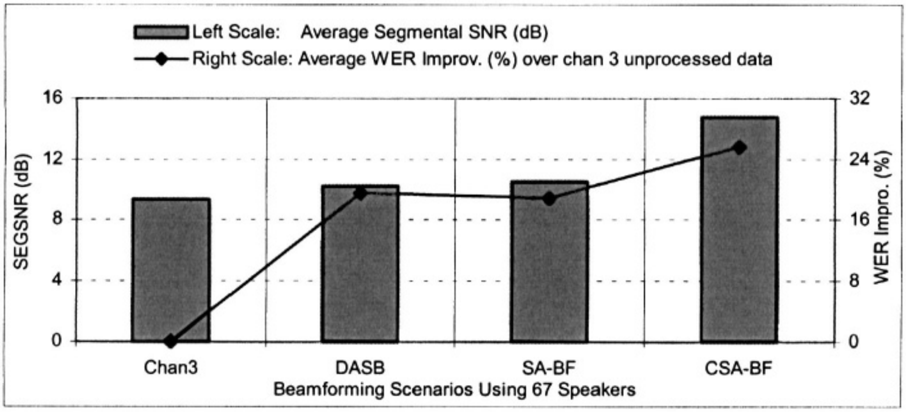


Figure 2-5. SEGSNR and WER Results for Reference Channel 3 Microphone (Chan3) and Array Processing/Beamforming Scenarios using 67 speakers from the CU-Move Corpus. Bar graph represents SegSNR in dB (using the left side scale), and line plot represents Avg. word error rate improvement (in %) (using the right side scale).

From these results (Table 2-1, Figure 2-5), we draw the following points:

1. Employing delay-and-sum beamforming (DASB) or the proposed speech adaptive beamforming (SA-BF), increases SegSNR slightly, but some variability exists across speakers. These two methods are able to improve WER for speech recognition by more than 19%.

2. There is a measurable increase in SegSNR and a decrease in WER when noise cancellation processing is activated (CSA-BF). With CSA-BF, SegSNR improvement is +5.5dB on the average, and also provides a relative WER improvement of 26%.

4.2 Environmental Sniffing

In this section we discuss our novel framework for extracting knowledge concerning environmental noise from an input audio sequence and organizing this knowledge for use by other speech systems. To date, most approaches dealing with environmental noise in speech systems are based on assumptions concerning the noise, or differences in collecting and training on a specific noise condition, rather than exploring the nature of the noise. We are interested in constructing a new speech framework which we have entitled *Environmental Sniffing* to detect, classify and track acoustic environmental conditions in the car environment (Figure 2-6, see [24,32]). The first goal of the framework is to seek out detailed information about the environmental characteristics instead of just detecting environmental changes. The second goal is to organize this knowledge in an effective manner to allow smart decisions to direct other speech systems. Our framework uses a number of speech processing modules including the Teager Energy Operator (TEO) and a hybrid algorithm with T^2-BIC segmentation, noise language modeling and broad class monophone recognition in noise knowledge estimation. We define a new information criterion, *Critical Performance Rate* (CPR), that incorporates the impact of noise into Environmental Sniffing performance by weighting the rate of each error type with a normalized cost function. We use an in-vehicle speech and noise environment as a test platform for our evaluations and investigate the integration of Environmental Sniffing into an Automatic Speech Recognition (ASR) engine in this environment.

We evaluate the performance of our framework using an in-vehicle noise database of 3 hours collected in 6 experimental runs using the same route and the same vehicle on different days and hours. Fifteen noise classes are transcribed during the data collection by a transcriber sitting in the car. The time tags are generated instantly by the transcriber. After data collection, some noise conditions are grouped together, resulting in 8 acoustically distinguishable noise classes.

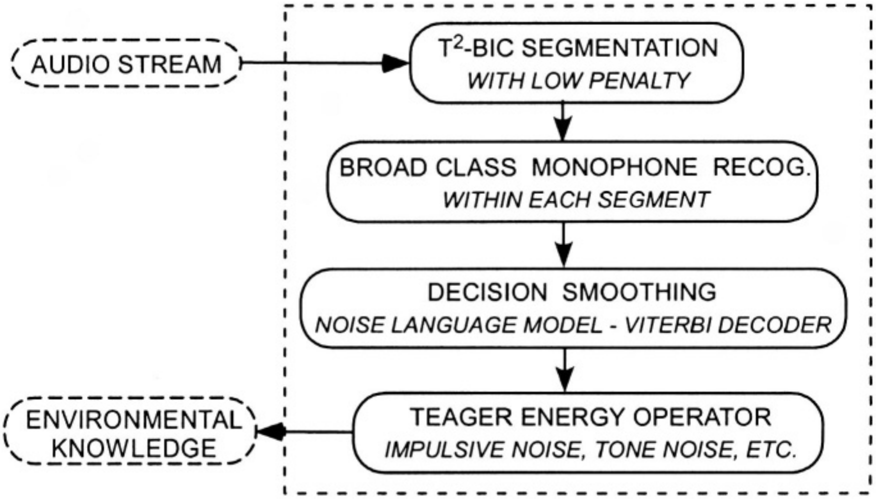


Figure 2-6. Flow Diagram for In-Vehicle Environmental Sniffing

We identified the following primary noise conditions of interest: (N1- idle noise consisting of the engine running with no movement and windows closed, N2- city driving without traffic and windows closed, N3- city driving with traffic and windows closed, N4- highway driving with windows closed, N5-highway driving with windows 2 inches open, N6- highway driving with windows half-way down, N7- windows 2 inches open in city traffic, NX-others), which are considered as long term acoustic environmental conditions. Other acoustic conditions (idle position with air-conditioning on, etc.) are matched to these primary classes having the closest acoustic characteristic.

Since the Environmental Sniffing framework is not a speech system itself, and must work with other speech systems, noise knowledge detection performance for each noise type should be calculated by weighting each classification error type by a term which is conditioned on the importance that error type plays in the subsequent speech application employing Environmental Sniffing. In [32], we specialized the formulation of CPR to a specific case where Environmental Sniffing framework is used for model selection within an ASR system. The Environmental Sniffing framework determines the initial acoustic model to be used according to the environmental knowledge it extracts. The knowledge in this context, will consist of the acoustic condition types with time tags. For this task, we can formulate the Critical Performance Rate as:

$$CPR = 1 - diag\{C \cdot \varepsilon^T\} \cdot \vec{a}^T, \tag{6}$$

where $\boldsymbol{\varepsilon}^{\mathbf{T}}$ denotes the transposed error matrix for noise classification, and **C** is the normalized cost matrix. Since some noise conditions occur more frequently than others, each noise condition will have an *a priori* probability denoted as **a.** Each cost value is proportional with WER difference between the matched case and the mismatched case, which is the performance deviation of the ASR engine by using the wrong acoustic model during decoding instead of using the correct acoustic model. The goal, in terms of performance, is to optimize the critical performance rate rather than optimizing the environmental noise classification performance rate, since it is more important to detect and classify noise conditions that have a more significant impact on ASR performance.

In our evaluations, we degraded the TI-DIGIT database at random SNR values ranging from -5 dB to +5 dB (i.e., -5,-3,-1,+1,+3,+5 dB SNR) with 8 different in-vehicle noise conditions using the noise database from [24]. A 2.5-hour noise data set was used to degrade the training set of 4000 utterances, and the 0.5 hour set was used to degrade the test set of 500 utterances (i.e., open noise degrading condition). Each digit utterance was degraded with only one acoustic noise condition.

Using the sniffing framework presented in Figure 2-6, each utterance was assigned to an acoustic condition. Using the fact that there was only one acoustic condition within each utterance, the Environmental Sniffing framework did not allow noise transitions within an utterance. A noise classification rate of 82% was obtained. Environmental condition specific acoustic models were trained and used during recognition tests. The Cost matrix **C** is calculated by testing different acoustic conditions using different acoustic models. The overall critical performance rate (CPR from Eq. (6)) was calculated as 92.1%

Having established the environmental sniffer, and normalized cost matrix for directing ASR model selection, we now turn to ASR system evaluation. We tested and compared the following 3 system configurations: S1-model matching was done using *a priori* knowledge of the acoustic noise condition (i.e., establish theoretical best performance – matched noise conditions), S2-model matching was done based on the environmental acoustic knowledge extracted from Environmental Sniffing, S3-all acoustic condition dependent models were used in a parallel multi-recognizer structure (e.g., ROVER) without using any noise knowledge and the recognizer hypothesis with the highest path score was selected.

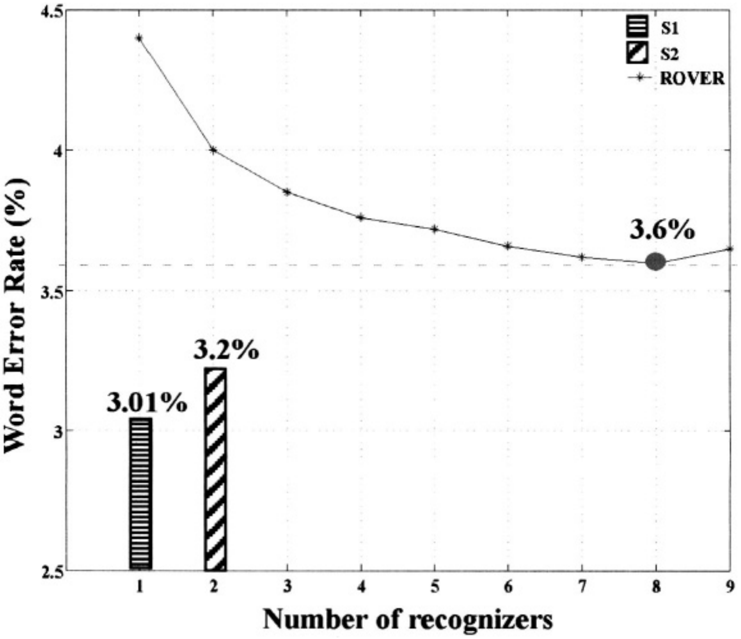


Figure 2-7. Word Error Rates for Digit Recognition Tests: S1 – matched noise model case, S2 – environmental sniffing model selection (1 CPU for sniffing, 1 CPU for ASR), S3 (ROVER) – employs up to 9 recognizers (i.e., CPUs) trained for each noise condition with ROVER selection.

As Figure 2-7 shows, system S1 achieved the lowest WER (i.e., 3.01%) since the models were matched perfectly to the acoustic condition during decoding. The WER for S2 was 3.2% using 2 CPU's (1 CPU for digit recognition, 1 CPU for sniffing acoustic conditions), which was close to the expected value of 3.23% (Note: in Figure 2-7, we plot system S2 with 2 CPU's even though only 1 ASR engine was used). S3 achieved a WER of 3.6% by using 8 CPU's. When we compare S2 and S3, we see that a relative 11.1% WER improvement was achieved, while requiring a relative 75% **reduction** in CPU resources. These results confirm the advantage of using Environmental Sniffing over an ASR ROVER paradigm.

There are two critical points to consider when integrating Environmental Sniffing into a speech task. First, and the most important, is to set up a configuration such as S1 where prior noise knowledge can be fully used to yield the lowest WER (i.e., matched noise scenario). This will require an understanding of the sources of errors and finding specific solutions assuming that there is prior acoustic knowledge. For example, knowing which speech enhancement scheme or model adaptation scheme is best for a specific acoustic condition is required. Secondly, a reliable cost matrix should be provided to the Environmental Sniffing so the subsequent speech task can calculate the expected performance in making an informed adjustment in the trade-off between performance and computation. For our experiments, we considered evaluation results for Environmental Sniffing where it is employed to find the *highest* possible acoustic condition so that the correct acoustic dependent model could be used. This is most appropriate for the goal of determining a single solution for the speech task problem at hand. If the expected performance for the system employing Environmental Sniffing is lower than the performance of a ROVER system, it may be useful to find the *n* most probable acoustic condition types among N acoustic conditions. In the worst case, the acoustic condition knowledge extracted from Environmental Sniffing could be ignored and the system will reduce to the traditional ROVER solution. The goal therefore in this section has been to emphasize that direct estimation of environmental conditions should provide important information to tailor a more effective solution to robust speech recognition systems.

4.3 Robust Speech Recognition

The CU-Move system incorporates a number of advances in robust speech recognition including a new more robust acoustic feature representation and built-in speaker normalization. Here, we report results from evaluations using CU-Move Release 1.1 A data from the extended digits part aimed at phone dialing applications.

Capturing the *vocal tract transfer function* (VTTF) from the speech signal while eliminating other extraneous information, such as speaker dependent characteristics and pitch harmonics, is a key requirement for robust and accurate speech recognition [33, 34]. The vocal tract transfer function is mainly encoded in the *short-term spectral envelope* [35]. Traditional MFCCs use the *gross spectrum* obtained as the output of a non-linearly spaced filterbank to represent the spectral envelope. While this approach is good for unvoiced sounds, there is a substantial mismatch for voiced and mixed sounds [34]. For voiced speech, the formant frequencies are biased towards strong harmonics and their bandwidths are misestimated [34,35]. MFCCs are known to be fragile in noisy conditions, requiring additional compensation for acceptable performance in realistic environments [45,28].

Minimum Variance Distortionless Response (MVDR) spectrum has a long history in signal processing but recently applied successfully to speech modeling [36]. It has many desired characteristics for a spectral envelope estimation method, most important being the fact it estimates the spectral powers accurately at the *perceptually important harmonics,* thereby providing an *upper envelope* which has strong implications for robustness in additive noise. Since the upper envelope relies on the high-energy portions of the spectrum, it will not be affected substantially by additive noise. Therefore, using MVDR for spectral envelope estimation for robust speech recognition is feasible and useful [37].

4.3.1 MVDR Spectral Envelope Estimation:

For details of MVDR spectrum estimation and its previous uses for speech parameterization, we refer the reader to [36,37,38,39,40]. In the MVDR spectrum estimation, the signal power at a frequency, ω_l, is determined by filtering the signal by a specially designed FIR filter, h(n), and measuring the power at its output. The FIR filter, h(n), is designed to minimize its output power subject to the constraint that its response at the frequency of interest, ωl, has unity gain. This constrained optimization is a key aspect of the MVDR method that allows it to provide a lower bias with a smaller filter length than the Periodogram method [41]. The Mth order MVDR spectrum can be parametrically written as;

$$P_{MV}(\omega) = \frac{1}{\sum_{k=-M}^{M} \mu(k) e^{-j\omega k}} = \frac{1}{\left|B(e^{j\omega})\right|^2} \tag{7}$$

The parameters, $\mu(k)$, can be obtained using the linear prediction (LP) coefficients, a_k, and the prediction error variance P_e [41].

$$\mu(k) = \begin{cases} \frac{1}{P_e} \sum_{i=0}^{M-k} (M+1-k-2i) a_i a_{i+k}^*, & k = 0,\ldots,M \\ \mu^*(-k) & k = -M,\ldots,-1 \end{cases} \tag{8}$$

4.3.2 Direct Warping of FFT Spectrum

The aim of using a non-linearly spaced filterbank is to remove the harmonic information that exists in voiced speech and smooth out the spectrum. MVDR, on the other hand, can handle voiced speech by accurately modeling spectral powers at the perceptually important harmonics. Therefore, it is both *useful* and *safe* to remove the filterbank structure and incorporate the perceptual considerations by directly warping the FFT spectrum. The warping can be incorporated via a first order all pass system [42]. In fact, both Mel and Bark scales can be implemented by changing only one system parameter, α. We use the phase response of the first order system in Eq. (9) as the warping function given in Eq. (10),

$$H(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \quad |\alpha| < 1 \tag{9}$$

$$\hat{\omega} = \tan^{-1} \frac{(1-\alpha^2)\sin\omega}{(1+\alpha^2)\cos\omega - 2\alpha} \tag{10}$$

where α determines the degree of warping. For 16kHz sampled signals, α=0.42 and α=0.55 approximates the Mel and Bark scales, respectively.

4.3.3 PMVDR Algorithm

We can summarize the PMVDR algorithm as follows [37];

- *Step 1:* Obtain the perceptually warped FFT power spectrum,
- *Step 2:* Compute the "perceptual autocorrelations" by using IFFT on the warped spectrum,
- *Step 3:* Perform an Mth order LP analysis via Levinson-Durbin recursion using perceptual autocorrelation lags [41],
- *Step 4:* Calculate the Mth order MVDR spectrum using Eq. (7) from LP coefficients [36],
- *Step 5:* Obtain Cepstrum coefficients using the straightforward FFT-based approach [43].

A flow diagram for the PMVDR algorithm is given in Figure 2-8. The algorithm is integrated into the CU-Move recognizer as the *default acoustic feature front-end,* (further information and code can be obtained from the CU-Move web site [27]).

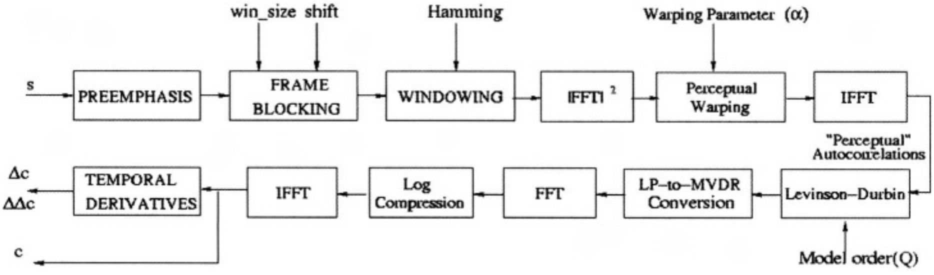


Figure 2-8. Flow Diagram of the PMVDR acoustic feature front-end

4.3.4 Experimental Evaluation

We evaluate the performance of PMVDR on the CU-Move extended digit task [27,28,37] using our SONIC [23,25] LVCSR system. Sonic incorporates speaker adaptation and normalization methods such as Maximum Likelihood Linear Regression (MLLR), Vocal Tract Length Normalization (VTLN), and cepstral mean & variance normalization. In addition advanced language-modeling strategies such as concept language models are also incorporated into the toolkit.

The training set includes 60 speakers balanced by age and gender, whereas the test set employs 50 speakers which again are age and gender-balanced. The word error rates (WER) and relative improvements of PMVDR with respect to MFCC are summarized in Table 2-2.

Gender/Sys.	**MFCC**	**PMVDR**	**Rel. Imp [%]**
Female	9.16	5.57	39.2
Male	13.22	8.76	33.7
Overall	**11.12**	**7.11**	**36.1%**

Table 2-2. WERs[%] and relative improvements for CU-Move task

The optimal settings for this task were found to be $M = 24$ and $\alpha = 0.57$ (close to the Bark scale). The 36.1% reduction in error rate using PMVDR features is a strong indicator of the robustness of these features in realistic noisy environments. We tested these features on a number of other tasks including clean, telephone and stressed speech and consistently obtain better results than that for MFCCs. Therefore, we conclude that PMVDR is a better acoustic front-end than MFCC for ASR in car environments.

4.3.5 Integration of Vocal Tract Length Normalization (VTLN)

VTLN is a well-known method of speaker normalization in which a customized linear warping function in the form of $\hat{f} = \beta f$ in frequency domain is used for each speaker [43]. The normalization factor, β, is a number which is generally less than 1.0 for female speakers and more than 1.0 for male speakers to account for different average vocal tract lengths. The normalization factor is determined by an exhaustive search as the one maximizing the total likelihood of a speaker's data using specifically trained models containing only 1 Gaussian for each phoneme cluster for a decision-tree state clustered HMM setting. The VTLN integrated with PMVDR requires *two consecutive warpings;* one for VTLN and one for incorporation of perceptual considerations.

Gender/Sys.	**No VTLN**	**VTLN**	**BISN**
Female	5.57	4.08	4.25
Male	8.76	7.17	7.10
Overall	**7.11**	**5.57**	**5.62**

Table 2-3. WERs [%] for Speaker normalization performance on CU-Move Corpus

In the PMVDR formulation, we used a first order system to perform perceptual warping. This warping function can also be used for speaker normalization in which the system parameter is adjusted to each speaker [44]. Rather than performing two consecutive warpings, we could simply change the degree of warping, (i.e., α), specifically for every speaker. This will enable us to perform both VTLN and perceptual warping using a *single warp.* The estimation of the VTLN-normalizing α can be done the same way as β. Such an integration of VTLN into the PMVDR framework yields an *acoustic front-end with built-in speaker normalization* (BISN). Table 2-3 summarizes our results with the conventional VTLN and BISN in the PMVDR framework.

The BISN yields comparable results to VTLN with a less complex front-end structure hence is an applicable speaker normalization method in ASR. The total WER reduction compared to the MFCC baseline is around 50% using PMVDR with BISN. The average warping factor for females was α_f=0.55 and for males α_m=0.59. Females require less warping than males due to shorter vocal tract length which conforms to VTLN literature.

Finally, experiments here were conducted on raw speech obtained from one microphone in our array. Using array processing techniques discussed in Sec. 4.1 and integrating the noise information obtained using techniques discusses in Sec. 4.2 will boost performance considerably when used in cascade with the robust acoustic front-end (PMVDR) and built-in speaker normalization (BISN). It is also possible and feasible to apply noise adaptation techniques such as Jacobian adaptation and speaker adaptation techniques such as MLLR to further improve performance[28]. Front-end speech enhancement schemes before acoustic feature extraction was also found to be useful in improving performance [28].

4.4 Proto-type Navigation Dialogue

Finally, we have developed a prototype dialog system for data collection in the car environment [46]. The dialog system is based on the DARPA Galaxy Communicator architecture [47,49] with base system components derived from the CU Communicator system [1,17]. Users interacting with the dialog system can enter their origin and destination address by voice. Currently, 1107 street names for Boulder, Colorado area are modeled. The dialog system automatically retrieves the driving instructions from the internet using an online WWW route direction provider. Once downloaded, the driving directions are queried locally from an SQL database. During interaction, users mark their location on the route by providing spoken odometer readings. Odometer readings are needed since GPS information has not yet been integrated into the prototype dialog system. Given the odometer reading of the vehicle as an estimate of position, route information such as turn descriptions, distances, and summaries can be queried during travel (e.g., "What's my next turn", "How far is it", etc.).

The system uses the University of Colorado SONIC [23,25,48] speech recognizer along with the Phoenix Parser[1] for speech recognition and semantic parsing. The dialog manager is mixed-initiative and event driven [1,17]. For route guidance, the natural language generator formats the driving instructions before presentation to the user by the text-to-speech (TTS) server. For example, the direction, "Park Ave W. becomes 22nd St." is reformatted to, "Park Avenue West becomes Twenty Second Street". Here, knowledge of the task-domain can be used to significantly improve the quality of the output text. The TTS system is based on variable-unit concatenation of synthesis units. While words and phrases are typically concatenated to produce natural sounding speech, the system can back off to smaller units such as phonemes to produce unseen words.

5. DISCUSSION

In this study, we have considered the problem of formulating an in-vehicle speech dialogue system for route navigation and planning. We discussed a flow diagram for our proposed system, CU-Move, and presented results from several sub-tasks including development of our microphone array CSA-BF processing scheme, environmental sniffing, speech enhancement processing, robust PMVDR features with built-in vocal tract length normalization, and a proto-type dialogue interface via the WWW. We also discussed our speech data corpus development based on Phase I: In-Vehicle Acoustic Noise measurements and Phase II: speech/speaker dialogue collection. Clearly, a number of challenges exist in the development and integration of a natural interactive system in such diverse and changing acoustic conditions. We believe that the processing tasks and results presented reflect useful steps in both the formulation of the CU-Move speech system, as well as contributing to a better scientific understanding of how to formulate dialogue systems in such adverse conditions. Finally, while the prospect of natural hands-free dialog within car environments is a challenging task, we feel that true fundamental advances will only occur if each of the processing phases are capable of sharing knowledge and leveraging their individual contributions to achieve a reliable overall working system.

REFERENCES

[1] W. Ward, B. Pellom, "The CU Communicator System," Proc. IEEE Work. Auto. Speech Recog. & Under., Keystone Colorado, 1999.

[2] J.H.L. Hansen, M.A. Clements, "Constrained Iterative Speech Enhancement with Application to Speech Recognition," *IEEE Trans. Signal Processing,* **39**(4):795-805, 1991.

[3] B. Pellom, J.H.L. Hansen, "An Improved Constrained Iterative Speech Enhancement Algorithm for Colored Noise Environments," IEEE Trans. Speech & Audio Proc., **6**(6):573-79, 1998.

[4] P. Lockwood, J. Boudy, "Experiments with a Nonlinear Spectral Subtractor (NSS), HMMs and the projection, for robust speech recognition in cars," *Speech Communication,* **11**:215-228, 1992.

[5] S. Riis,O.Viikki, "Low Complexity Speaker Independent Command Word Recognition in Car Environments, IEEE ICASSP-00, **3**:1743-6, Istanbul, Turkey, 2000.

[6] J. Huang, Y. Zhao, S. Levinson, "A DCT-based Fast Enhancement Technique for Robust Speech Recognition in Automobile Usage," EUROSPEECH-99, **5**:1947 –50, Budapest, Hungary, 1999.

[7] R. Bippus, A. Fischer, V. Stahl, "Domain Adaptation for Robust Automatic Speech Recognition in Car Environments," EUROSPEECH-99, **5**:1943-6, Budapest, Hungary, 1999.

[8] A. Fischer, V. Stahl, "Database And Online Adaptation For Improved Speech Recognition In Car Environments," IEEE ICASSP-99, Phoenix, AZ, 1999.

[9] L.S. Huang, C.H. Yang, "A Novel Approach to Robust Speech Endpoint Detection in Car Environments," IEEE ICASSP-00, **3**:1751-4, Istanbul, Turkey, 2000.

[10] E.Ambikairajah, G. Tattersall, A. Davis, "Wavelet Transform-based Speech Enhancement," ICSLP-98, **7**:2811-14, Sydney, Australia, 1998.

[11] P. Gelin, J.-C. Junqua , “Techniques for Robust Speech Recognition in the Car Environment,” EUROSPEECH-99, **6**:2483-6, Budapest, Hungary, 1999.

[12] http://www.speechdat.com/SP-CAR/

[13] P. Pollák, J. Vopièka, P. Sovka, “Czech Language Database of Car Speech and Environmental Noise,” EUROSPEECH-99, **5**:2263-6, Budapest, Hungary, 1999.

[14] P. Geutner, M. Denecke, U. Meier, M. Westphal, A. Waibel, “Conversational Speech Systems For On-Board Car Navigation and Assistance,” ICSLP-98, paper #772, Sydney, Australia, 1998.

[15] M. Westphal, A. Waibel, “Towards Spontaneous Speech Recognition for On-Board Car Navigation and Information Systems,” EUROSPEECH-99, **5**: 1955-8, Budapest, Hungary, 1999.

[16] J.H.L. Hansen, “Analysis and Compensation of Speech under Stress and Noise for Environmental Robustness in Speech Recognition,” Speech Comm., pp 151-170, Nov. 1996.

[17] B. Pellom, W. Ward, S. Pradhan, “The CU Communicator: an Architecture for Dialogue Systems,” ICSLP-2000, Beijing, China, Oct. 2000.

[18] J.F. Kasier, “On a Simple Algorithm to Calculate the ‘Energy’ of a Signal”, IEEE ICASSP-90, pp. 381--384, 1990.

[19] X. Zhang, J.H.L. Hansen, “CSA-BF: Novel Constrained Switched Adaptive Beamforming for Speech Enhancement & Recognition in Real Car Environments”, IEEE ICASSP-03. pp. 125-128, Hong Kong, China, April 2003.

[20] P. L. Feintuch, N. J. Bershad, and F. A. Reed, “Time delay Estimation Using the LMS Adaptive Filter-Static Behavior”, *IEEE Trans. Acoustics, Speech, Signal Proc.,* ASSP-29(3):571--576, June 1981.

[21] J.H.L. Hansen, et.al., “CU-Move”: Analysis & Corpus Develop. for Interactive In-vehicle Speech Systems”, Eurospeech-01, pp. 2023-2026, Aalborg, Denmark, 2001.

[22] http://www.nist.gov

[23] Pellom, “Sonic: The University of Colorado Continuous Speech Recognizer”, *University of Colorado, Technical Report #TR-CSLR-2001-01,* Boulder, Colorado, March, 2001.

[24] M. Akbacak, J.H.L. Hansen, “Environmental Sniffing: Noise Knowledge Estimation for Robust Speech Systems,” IEEE ICASSP-2003, pp. 113-116, Hong Kong, China, April 2003.

[25] B. Pellom, K. Hacioglu, “Recent Improvements in the CU Sonic ASR System for Noisy Speech,” ICASSP-2003, Hong Kong, China, April 2003.

[26] J.H.L. Hansen, “Getting Started with the CU-Move Corpus”, Release 2.0A Technical Report, 44pgs., Nov. 17, 2002 [see http://cumove.colorado.edu/].

[27] http://cumove.colorado.edu/

[28] U. Yapanel, X. Zhang, J.H.L. Hansen, “High Performance Digit Recognition in Real Car Environments,” ICSLP-2002, vol. 2, pp. 793-796, Denver, CO.

[29] http://speechfind.colorado.edu/

[30] http://speechbot.research.compaq.com/

[31] J.H.L. Hansen, C. Swail, A.J. South, R.K. Moore, H. Steeneken, E.J. Cupples, T. Anderson, C.R.A. Vloeberghs, I. Trancoso, P. Verlinde, “The Impact of Speech Under ‘Stress’ on Military Speech Technology,” NATO RTO-TR-10, AC/323(IST)TP/5 IST/TG-01, March 2000.

[32] M. Akbacak, J.H.L. Hansen, “Environmental Sniffing: Robust Digit Recognition for an In-Vehicle Environment,” Eurospeech-03, pp. 2177-2180, Geneva, Switzerland, Sept. 2003.

[33] M. J. Hunt, “Spectral Signal Processing for ASR”, Proc ASRU’99, Keystone, Colorado , USA

[34] L. Gu and K. Rose, “Perceptual Harmonic Cepstral Coeffiecients as the Front-End for Speech Recognition”, ICSLP-00, Beijing, China, 2000.

[35] M. Jelinek and J.P. Adoul, “Frequency-domain Spectral Envelope Estimation for Low Rate Coding of Speech”, IEEE ICASSP-99, Phoenix, Arizona, 1999.

[36] M.N. Murthi and B.D. Rao, “All-pole Modeling of Speech Based on the Minimum Variance Distortionless Response Spectrum”, IEEE Trans. Speech & Audio Processing, May 2000.

[37] U.H. Yapanel and J.H.L. Hansen, “A New Perspective on Feature Extraction for Robust In-vehicle Speech Recognition”, Eurospeech-03, pp.1281-1284, Geneva, Switzerland, Sept. 2003.

[38] S. Dharanipragada and B.D. Rao, “MVDR-based Feature Extraction for Robust Speech Recognition”, IEEE ICASSP-01, Salt Lake City, Utah, 2001.

[39] U.H. Yapanel and S. Dharanipragada, “Perceptual MVDR-based Cepstral Coefficients for Noise Robust Speech Recognition”, IEEE ICASSP-03, Hong Kong, China, April 2003.

[40] U.H. Yapanel, S. Dharanipragada, J.H.L. Hansen, “Perceptual MVDR-based Cepstral Coefficients for High-accuracy Speech Recognition”, Eurospeech-03, pp. 1425-1428, Geneva, Switzerland, Sept. 2003.

[41] S.L. Marple, Jr, “Digital Spectral Analysis with Applications”, Prentice-Hall, Englewood Cliffs, NJ, 1987

[42] K. Tokuda, T. Masuko, T. Kobayashi, and S. Imai, “Mel-generalized Cepstral Analysis-A Unified Approach to Speech Spectral Estimation”, ICSLP-94, Yokohama, Japan, 1994.

[43] L.F. Uebel and P.C. Woodland, “An Investigation into Vocal Tract Length Normalization”, Eurospeech-99, Budapest, Hungary, 1999.

[44] J. McDonough, W. Byrne, and X. Luo, “Speaker Normalization with All-pass Transforms”, ICSLP-98, Sydney, Australia, 1998.

[45] S.E. Bou-Ghazale, J.H.L. Hansen, “A Comparative Study of Traditional and Newly Proposed Features for Recognition of Speech Under Stress,” IEEE Trans. Speech & Audio Proc., **8**(4): 429-442, July 2000.

[46] B. Pellom, W. Ward, J.H.L. Hansen, K. Hacioglu, J. Zhang, X. Yu, S. Pradhan, “University of Colorado Dialog Systems for Travel and Navigation”, in Human Language Technology Conference (HLT), San Diego, California, March, 2001.

[47] URL: Galaxy Communicator Software, http://communicator.sourceforge.net

[48] URL: University of Colorado SONIC LVCSR System http://cslr.colorado.edu/beginweb/speech_recognition/sonic.html

[49] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, V. Zue, “Galaxy-II: A Reference Architecture for Conversational System Development,” *Proc. ICSLP,* Sydney Australia, Vol. 3, pp. 931-934, 1998.

[50] X. Zhang, J.H.L. Hansen, “CSA-BF: Novel Constrained Switched Adaptive Beamforming for Speech Enhancement & Recognition in Real Car Environments”, IEEE Trans. on Speech & Audio Processing, vol. 11, pp. 733-745, Nov. 2003.

Chapter 3

A SPOKEN DIALOG CORPUS FOR CAR TELEMATICS SERVICES

Masahiko Tateishi[1], Katsushi Asami[1], Ichiro Akahori[1], Scott Judy[2], Yasunari Obuchi[3], Teruko Mitamura[2], Eric Nyberg[2], and Nobuo Hataoka[4]

[1]Research Laboratories, DENSO CORPORATION, 500-1, Minamiyama , Nisshin, Aichi, 470-0111, Japan; [2]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA; [3]Advanced Research Laboratory, Hitachi Ltd., 1-280, Higashi-koigakubo, Kokubunji, Tokyo, 185-8601, JAPAN; [4]Central Research Laboratory, Hitachi Ltd., 1-280, Higashi-koigakubo, Kokubunji, Tokyo, 185-8601, JAPAN Email: mtatei@rlab.denso.co.jp

Abstract: Spoken corpora provide a critical resource for research, development and evaluation of spoken dialog systems. This chapter describes the spoken dialog corpus used in the design of CAMMIA (Conversational Agent for Multimedia Mobile Information Access), which employs a novel dialog management system that allows users to switch dialog tasks in a flexible manner. The corpus for car telematics services was collected from 137 male and 113 female speakers. The age distribution of speakers is balanced in the five age brackets of 20's, 30's, 40's, 50's, and 60's. Analysis of the gathered dialogs reveals that the average number of dialog tasks per speaker was 8.1. The three most frequently-requested types of information in the corpus were traffic information, tourist attraction information, and restaurant information. Analysis of speaker utterances shows that the implied vocabulary size is approximately 5,000 words. The results are used for development and evaluation of automatic speech recognition (ASR) and dialog management software.

Keywords: Spoken Dialog Corpus, Telematics, Speech Recognition, Dialog Tasks.

1. INTRODUCTION

The term *telematics* refers to the emerging industry of communication, information, and entertainment services delivered to motor vehicles via wireless network technology. A telematics system must provide a human-machine interface (HMI) that allows drivers to operate the device, system or service easily and without any risks regarding traffic safety. A spoken dialog system is considered to be the most suitable HMI for telematics, since it allows the driver to keep “hands on the wheel, eyes on the road”.

The Conversational Agent for Multimedia Mobile Information Access (CAMMIA) provides a framework for client-server implementation of spoken dialog systems in mobile, hands-free environments[1][5]. The goal of CAMMIA is to realize large-scale speech dialog systems that can handle a variety of information retrieval tasks. CAMMIA is based on VoiceXML, a markup language for speech dialog systems which has been proposed as a standard by W3C [7]. The client is an in-vehicle terminal with an automatic speech recognition (ASR) system, a VoiceXML interpreter, and a text-to-speech (TTS) system; the server is a separate computer which runs a Dialog Manager (DM) module [5]. The client recognizes the driver’s utterances according to the VoiceXML dialog scenarios, and transmits the recognition results in the form of requests to the server. The server then searches its database and transforms the search results into VoiceXML files which are transmitted to the client as a response.

One novel aspect of CAMMIA is the natural conversational interaction between the user and the system, supported by a DM module that allows the user to change dialog tasks flexibly. Many of the system requirements associated with natural spoken dialog can be ascertained by studying human behavior as observed in large collections of spoken or written data. Specifically, the analysis includes defining a lexicon and grammar for ASR, as well as designing suitable dialog scenarios for use by the DM.

Human-computer dialog differs from human-human dialog in various aspects, including linguistic complexity[2]. However, the examination of human-human dialogs is a natural first step in the process of modeling human dialog behavior [3]. The modeling approach requires very large quantities of task-oriented linguistic data. To meet this requirement, we collected a spoken dialog corpus for car telematics services. In this Chapter, Section 2 outlines the system architecture of CAMMIA. Section 3 explains the spoken dialog corpus collection. Section 4 describes the analysis of the corpus, followed by conclusions.

2. SYSTEM ARCHITECTURE

Figure 3-1 depicts the current system architecture of CAMMIA, which consists of in-vehicle terminals and the server which are connected by a wireless network. The client consists of an ASR, a VoiceXML interpreter, and a TTS, whereas the server consists of a DM, a database, and a set of dialog scenarios. The DM makes a VoiceXML text according to a dialog task using a Dialog Scenario database and delivers it to an in-vehicle terminal. In the in-vehicle terminal, the VoiceXML interpreter receives the VoiceXML text and real dialog interactions between a used and the system can be carried over. A spoken dialog corpus was used to evaluate the lexicon and grammar of ASR and suitable dialog scenarios that represent particular dialog tasks such as traffic information. The corpus was also used to evaluate the system's coverage of different dialog tasks; these evaluations are discussed in Section 5.

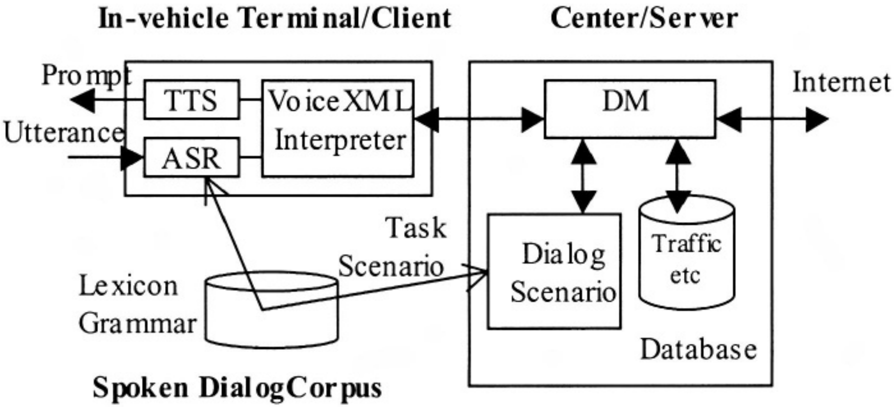


Figure 3-1. System Architecture of CAMMIA.

3. COLLECTION OF SPOKEN DIALOG CORPUS

The collection of a spoken dialog corpus consists of three steps: speaker selection, experimental setup, and task configuration. The most difficult and important thing is to encourage the speakers to interact spontaneously and effectively. The experimental setup and task configuration have a significant impact on their ability to do so. This section focuses on the experimental setup and task configuration which was employed.

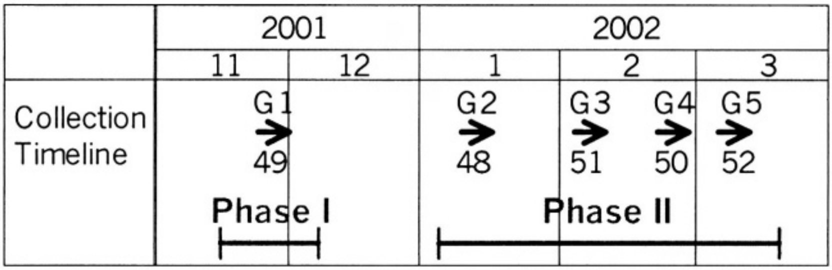


Figure 3-2. Collection timeline of spoken dialog corpus.

3.1 Speaker Selection and Collection Timeline for the Spoken Dialog Corpus

The nature of the dialog task and the lexicon and grammar which describe a speaker's utterances vary significantly according to the gender and age of the speaker. Therefore, it is desirable to balance the gender and age range of the speakers in the set of experimental subjects. We collected the spoken dialog corpus from 250 speakers, consisting of 137 males and 113 females. The age distribution of speakers was also balanced in the five age brackets of 20's, 30's, 40's, 50's, and 60's years old. All were residents of the Tokyo Metropolitan Area and 235 of them held driver's licenses; fifty of the subjects had prior experience with car navigation systems.

We divided the 250 speakers into five groups, G1 to G5, consisting of approximately 50 speakers per group. The spoken dialog corpus was collected from G1 to G5, in order, according to the timeline depicted in Figure 3-2. The numbers of speakers in G1 to G5 are also shown under the arrows in the figure. The number of speakers in each group differed because it was necessary to arrange the data collection according to the individual schedules of the subjects. After collecting data from each group, we improved the experimental setup and the task configuration before proceeding with the next group. The most significant improvements were introduced after collecting data from G1. Therefore, in this article we refer to the data collection from G1 as Phase I, and the data collection from G2 through G5 as Phase II. The next section discusses the difference of the experimental setup and task configuration between Phase I and II.

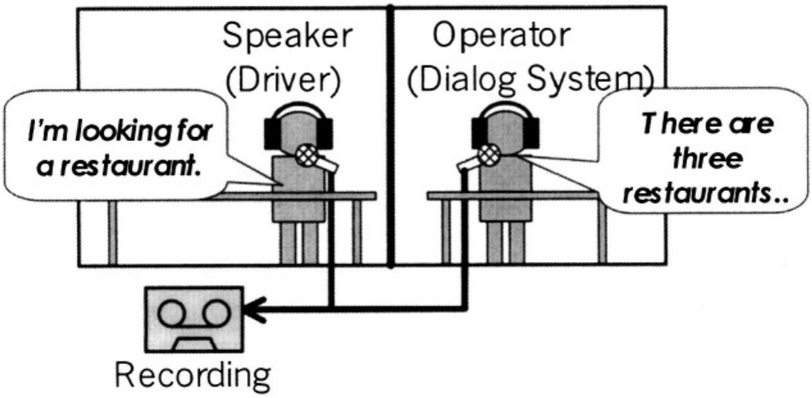


Figure 3-3. Experimental setup (Phase I).

3.2 Collection of Spoken Dialog Corpus – Phase I

3.2.1 The Experimental Setup

The spoken dialog corpus was recorded in a studio in Tokyo. The studio consisted of two rooms as depicted in Figure 3-3. The left room represented a car equipped with an in-vehicle terminal, whereas the right room represented the remote server. In the left room, the speaker assumed the role of driver, and in the right room, the operator assumed the role of the dialog system. The speaker and the operator talked to each other using connected microphones and headsets; the two rooms were completely separated so that no nonverbal interactions took place between speaker and operator. The dialogs that took place between the speaker and the operator were recorded on audiotape. The operator answered queries from the speaker by acting as a travel agent with access to the following information:

- Real-time traffic information
- Restaurant information
- Sightseeing information
- Hotel and *ryokan* (Japanese inn) information

In order to support task-oriented dialogs including detailed tourist information, the operator role required expertise in travel information in addition to the ability to speak correct Japanese. To meet these requirements, we employed a professional female announcer with prior job experience in a travel agency to act in the role of operator.

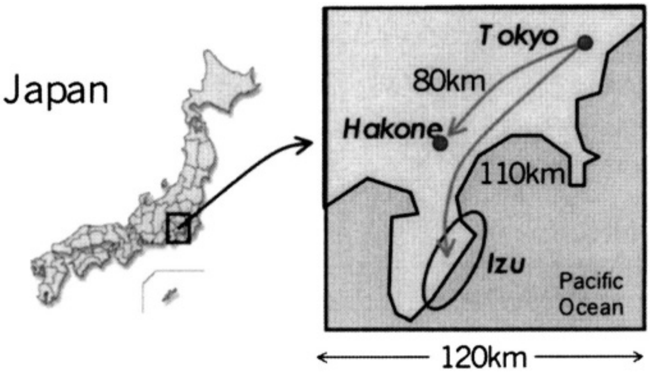


Figure 3-4. Sightseeing areas employed in the task (Phase I and II).

3.2.2 The Task and Instructions

A preliminary survey of 302 drivers indicated that the three most desirable types of driving information are traffic information, restaurant information, and sightseeing information. As a result, data was gathered for three predefined dialog tasks:

Task 1: Obtain route guidance to a particular destination.
Task 2: Find a restaurant for lunch.
Task 3: Find tourist attractions to visit after lunch.

The speaker was instructed to pretend that he/she was in a car equipped with an in-vehicle terminal, traveling on an overnight trip to one of two sightseeing areas outside Tokyo. The number of possible destinations was limited to two because it was difficult for the operator to prepare sufficient driving information for multiple sightseeing destinations in advance. The chosen destinations were chosen to be sufficiently popular to motivate the speakers to generate ample questions. In addition, the destinations were chosen to be not too far from Tokyo (destinations that are far from Tokyo require use of expressways, which renders route guidance less useful).

Keeping these constraints in mind, we selected Hakone and Izu as the possible tourist destinations (see Figure 3-4). Both of these sightseeing areas have many hot springs and tourist attractions, and are very popular to residents in the Tokyo Metropolitan Area. There are several ways to reach each of these areas from Tokyo.

The speaker was instructed to talk with the operator and obtain appropriate driving information for Tasks 1 through 3 listed above. Each task was printed in a handout, and each speaker utilized this handout during dialogs with the operator.

3.3 Collection of Spoken Dialog Corpus – Phase II

This section describes the improvements we introduced to the collection of spoken dialog corpus after Phase I.

3.3.1 The Task and Instructions

Phase I collection revealed several problems in the task and instructions. The first problem stemmed from the use of handouts. The speakers tended to recite the texts from the handout verbatim when they initiated a dialog. For example, suppose the handout read "Task 2: You've just arrived in Hakone. Find a restaurant for lunch." If the operator started the dialog by saying "Driving information center. May I help you?", the speaker might respond with "Uh, I've just arrived in Hakone, Please find a restaurant for lunch." This phenomenon prevented us from collecting spontaneous speech samples in some cases.

The second problem was that the predefined tasks did not adequately encourage the speakers to pretend that they were on a real trip, such that they failed to generate questions relevant to the tasks. As a result, the operator sometimes had to halt the conversation and instruct the speakers regarding the types of questions they could ask to make the dialog more realistic.

To address these initial problems, we divided the remaining 201 speakers into 40 groups, each of which consisted of five or six speakers. Each group was instructed to choose Hakone or Izu as their destination, and to discuss a driving plan for an overnight trip according to their interests. After the discussion, each speaker generated two sets of dialog tasks, *A* and *B,* relevant to the driving plan. Set *A* and set *B* listed the questions to obtain information required before starting the trip on the first day, and before leaving the hotel on the second day, respectively (see Figure 3-5). The recording of the dialog was also divided into two sessions, *A* and *B* which corresponded to the first day and the second day, respectively.

In addition, we found that it was necessary to provide the speaker with additional details for the task, such as the date of the trip and the travel expense limit. Road congestion was varied based on a distinction between weekday and weekend travel, and the operator altered the route guidance accordingly. The specified dtravel expense limit forced the speaker to construct a reasonable travel plan. In order to avoid the previously mentioned problems associated with providing these details on a text handout, we used concrete physical objects to represent the travel details. For example, a calendar with a particular day marked was used to indicate the day of the trip, and a wallet containing banknotes was used to indicate the travel expense limit.

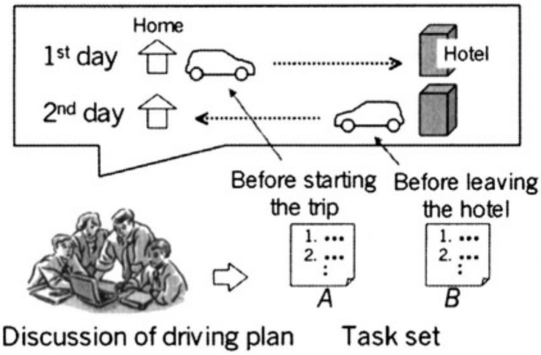


Figure 3-5. The task and instructions (Phase II).

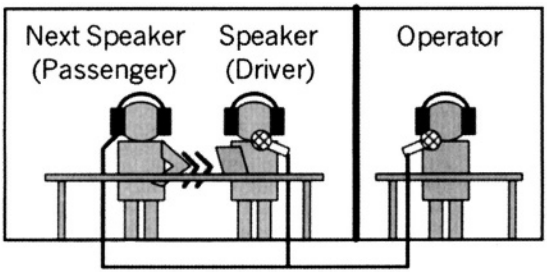


Figure 3-6. The experimental setup (Phase II).

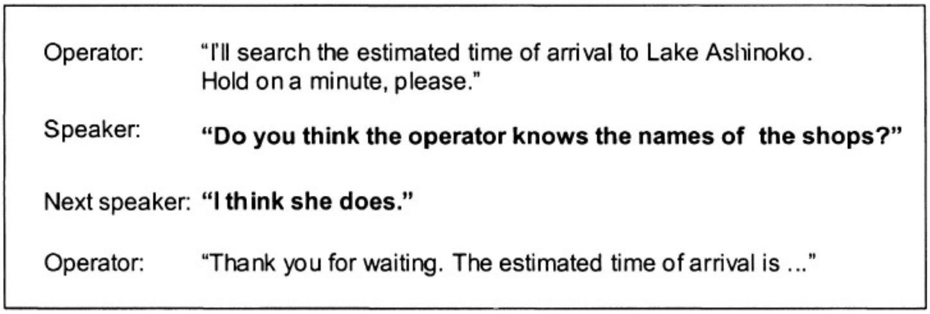
3.3.2 The Experimental Setup

We also improved the experimental setup following Phase I. In order to let the next speaker warm up and to ease nervousness, we had the next speaker sit beside the current speaker and act as a passenger, as depicted in Figure 3-6. The next speaker can then monitor the conversation between the speaker and the operator.

We found that introducing the next speaker first as a passenger had an additional effect of eliciting additional questions from the current speaker; for example:

a) The next speaker nudged the current speaker when the current speaker forgot to ask a pertinent question;
b) When the operator paused the dialog to search for requested information, the current speaker and the next speaker conferred regarding what to ask next, as shown in Figure 3-7.

These improvements helped the current speaker to generate appropriate questions more easily. The operator commented that the speakers in Phase II were more proactive than Phase I. Analysis of the corpus revealed that the initial 49 speakers in Phase I required 68 instruction explanations from the operator, whereas the following 201 speakers in Phase II required only one instruction explanation.



Operator: "I'll search the estimated time of arrival to Lake Ashinoko. Hold on a minute, please."

Speaker: **"Do you think the operator knows the names of the shops?"**

Next speaker: **"I think she does."**

Operator: "Thank you for waiting. The estimated time of arrival is ..."

Figure 3-7. Example discussion between the current speaker and the next speaker (Phase II).

4. TRANSCRIPTION AND TAGGING OF SPOKEN DIALOG CORPUS

Each dialogue was transcribed into text by hand from audio tape. Each dialog was segmented into individual utterances; the prefixes 'L:' and 'R:' were used to label operator and speakers utterances, respectively.

Proper nouns in the text were annotated using the bracketted format {*X* name}, where *X* is one of the 5 letters **{A, P, R, S, W}** representing tourist attractions, places, railway facilities, shops/restaurants, and traffic facilities, respectively. Traffic facilities include the names of roads, entrances/exits of expressways, etc. Annotating proper nouns is useful for designing the lexicon and grammar for ASR. It is also convenient for designing class *N*-grams used by the ASR, where proper nouns with the label *X* form a class *X*.

Next, we tagged each of the speaker utterances using the labels to distinguish the type of task to which they belonged. We defined the following ten task labels:

- Introduction
- Traffic
- Restaurant
- Tourist attraction
- Shopping
- Hotel
- Parking
- Weather
- Facility
- General

The 'Introduction' label indicates utterances belonging to the introductory part of the next task, which is typically attested at the beginning of the dialog and explains the speaker's intent to the operator.

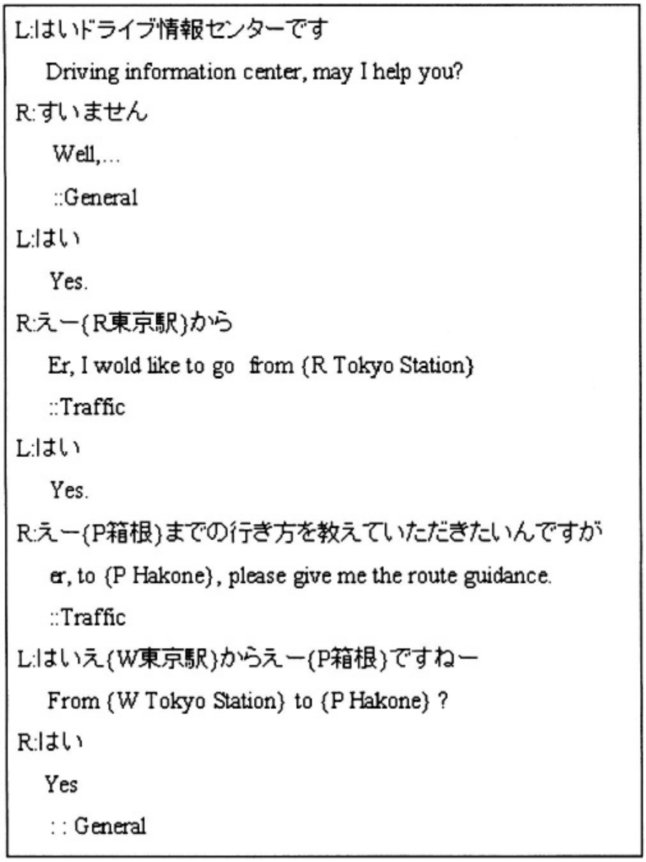
The 'Parking' label indicates utterances belonging to the task of locating parking lot information. However, we do not label utterances as 'Parking' when the speaker is talking about parking lots for other facilities, such as restaurants and tourist attractions. For example, the utterance 'Does the restaurant have a parking lot?' would be labeled as 'Restaurant'.

The 'Facility' label indicates utterances belonging to information about facilities such as gas stations, ATMs, toilets, etc.

The 'General' label indicates utterances that may appear in any type of task. For example, "Yes." and "I see." are labeled as 'General'.

Figure 8 shows an example of a transcribed text. English translations appear below the utterances. The annotated proper nouns are "Tokyo Station" and "Hakone". The task labels are shown using the notation '::<Task label>'. The dialog starts with the operator utterance "Driving information center, may I help you?" The speaker desires route guidance from Tokyo Station to Hakone. The speaker's second and third utterances are labeled as 'Traffic'. Other speaker utterances, 'Yes.' and 'Well,…' are labeled as 'General'.

Figure 3-9 shows examples of utterances labeled as 'Introduction'. With these utterances, the speaker explains that she is going to Izu for the first time before asking about hotel information.



L:はいドライブ情報センターです

Driving information center, may I help you?

R:すいません

Well,...

::General

L:はい

Yes.

R:えー{R東京駅}から

Er, I wold like to go from {R Tokyo Station}

::Traffic

L:はい

Yes.

R:えー{P箱根}までの行き方を教えていただきたいんですが

er, to {P Hakone}, please give me the route guidance.

::Traffic

L:はいえ{W東京駅}からえー{P箱根}ですねー

From {W Tokyo Station} to {P Hakone} ?

R:はい

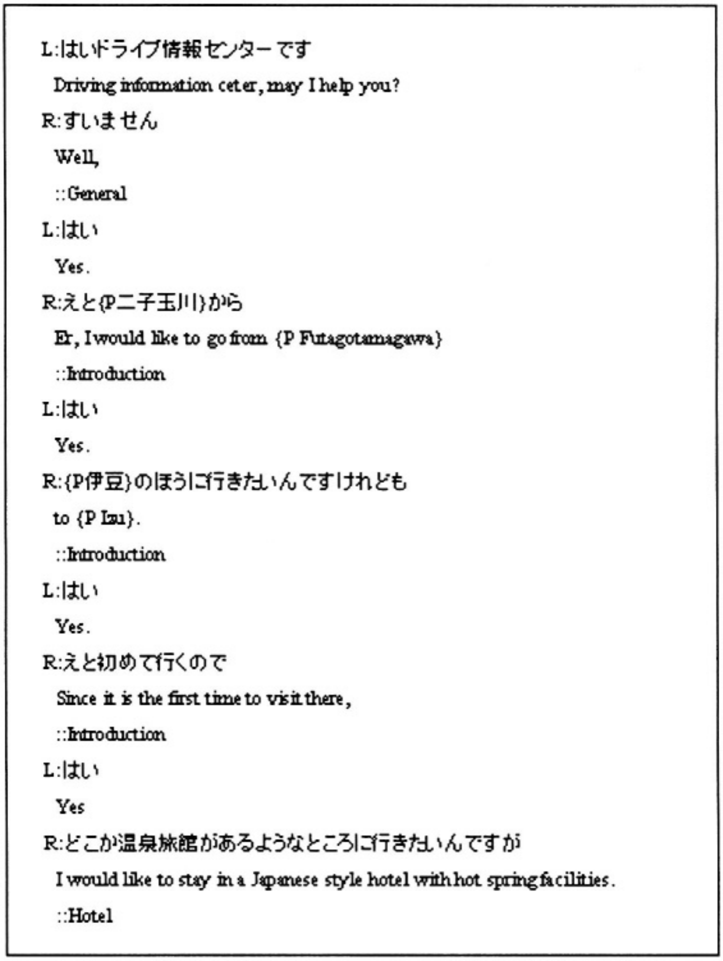
Yes

: : General

Figure 3-8. Example of Transcribed Text.

5. ANALYSIS OF THE SPOKEN DIALOG CORPUS

The spoken dialog corpus plays several roles in the design of CAMMIA: a) as a reference for creating the ASR grammar; b) as a test suite for testing the ASR grammar; c) as a resource for lexicon development; d) as a guide for identifying the highest frequency sentences and words; and e) as a reference for possible task scenarios.



L:はいドライブ情報センターです
Driving information ceter, may I help you?
R:すいません
Well,
::General
L:はい
Yes.
R:えと{P二子玉川}から
Er, I would like to go from {P Futagotamagawa}
::Introduction
L:はい
Yes.
R:{P伊豆}のほうに行きたいんですけれども
to {P Izu}.
::Introduction
L:はい
Yes.
R:えと初めて行くので
Since it is the first time to visit there,
::Introduction
L:はい
Yes
R:どこか温泉旅館があるようなところに行きたいんですが
I would like to stay in a Japanese style hotel with hot spring facilities.
::Hotel

Figure 3-9. Examples of utterances labeled as 'Introduction'.

The spoken dialog corpus consists of 450 conversations comprising 34,612 operator utterances and 33,773 speaker utterances. As described in Section 3, we improved the task configuration, instruction and experimental setup significantly in Phase II. Therefore, we focus on the analysis of the corpus acquired in Phase II. In this section, we discuss the statistical analysis of speaker utterances to develop a lexicon for ASR. We also discuss the types of individual tasks attested in the corpus.

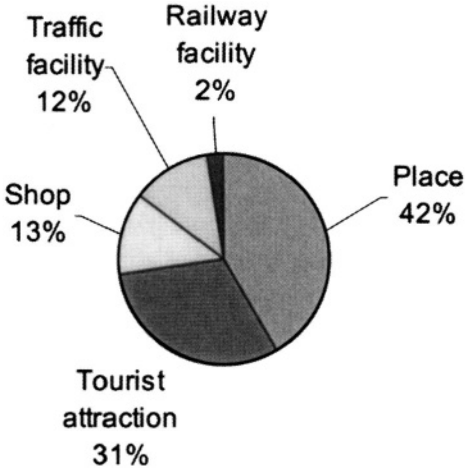


Figure 3-10. Ratio of vocabulary size of proper nouns.

5.1 Statistics of Speakers Utterances

The number of speakers in Phase II was 201. Six of the speakers misunderstood the instructions, and the corresponding dialogs were removed from the corpus. We analyzed the remaining 195 dialogs, which are referred to as the "spoken dialog corpus of Phase II" in the following discussion. The corpus consisted of 390 conversations which included 28,334 operator utterances and 27,509 speaker utterances.

The spoken dialog corpus was segmented into morphemes using ChaSen[4], a Japanese morphological analyzer. The vocabulary size for speakers' utterances was 4,533 words, consisting of 762 proper nouns and 3,771 words other than proper nouns. The set of proper nouns varied according to the sightseeing area selected. The remaining words were more general to the overall set of dialog tasks. From these observations, we concluded that the lexicon for ASR would be approximately 4,000 to 5,000 words to support recognition of speaker utterances for this general family of tasks.

Figure 3-10 provides a categorization of annotated proper nouns. Four types of proper nouns (places, tourist attractions, shops and traffic facilities) comprised about 98% of the proper nouns attested in the corpus.

5.2 Statistics of Tasks

In Phase I, we specified a sequence of three tasks: 1) traffic; 2) restaurant; and 3) tourist attraction. In Phase II, we did not specify and ordered set of tasks. Speakers could switch dialog tasks freely during their interactions with the operator. This is a less restrictive approach for corpus collection, and a more varied set of dialog tasks appear in Phase II. Therefore, it is of much interest to analyze the number and types of dialog tasks as well as the task transitions attested in the corpus collected from Phase II.

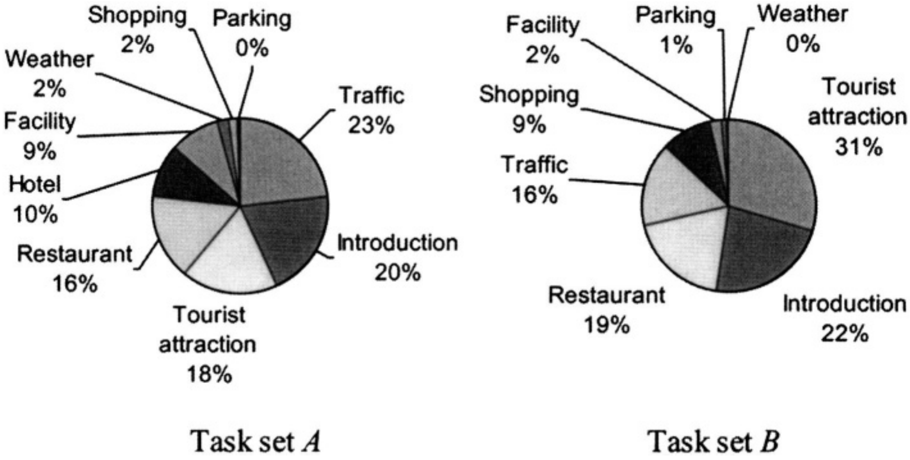
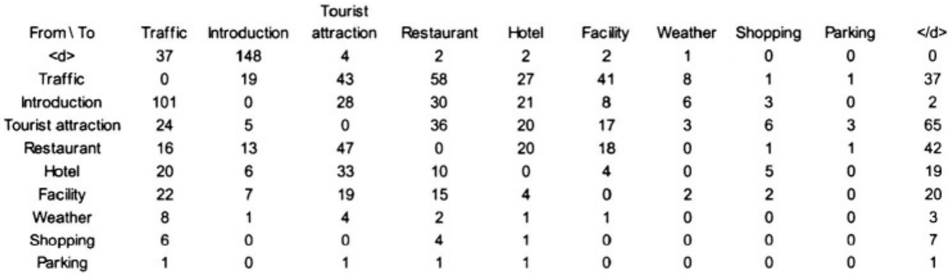


Figure 3-11. Task frequency ration in the set A and B.

Analysis of the collected dialogs revealed that the total number of dialog tasks in Phase II was 1,593, indicating an average number of tasks per speaker of 8.1. As described in Section 3.3, speakers in Phase II generated two sets of dialog task sets (*A* and *B)* according to their interests. These two sets corresponded to the task before starting the trip on the first day, and the task before leaving the hotel on the second day, respectively. The total number of tasks in sets *A* and *B* were 1,002 and 591, respectively, indicating that the average number of tasks per set per speaker was 5.1 and 3.0, respectively.

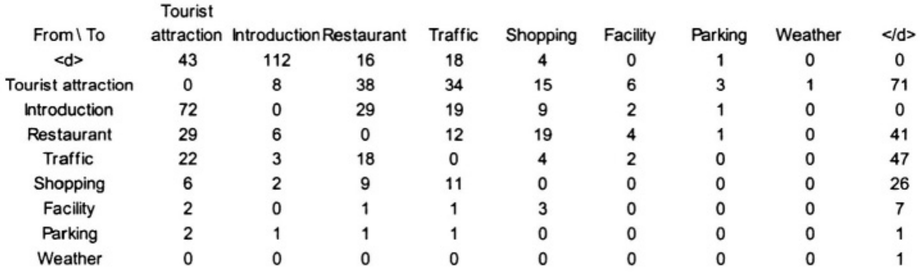
Figure 3-11 contains a comparison of the task frequency ratios in the set *A* and *B.* The three most desired types of information in the corpus were traffic information, tourist attraction information, and restaurant information. Naturally, the desire for hotel information disappeared in task set *B.* The most frequently requested information (task) on the first day was traffic information. On the other hand, the most frequently requested information on the second day was tourist attraction information. This suggests that speakers are most interested in traffic information before starting a trip, and tourist attraction information before leaving the hotel, which coincides with the intuition of most drivers. It is also worthwhile to note that the frequency of the shopping task increased from 2% to 9% on the second day, since many speakers were interested in purchasing souvenirs on their way home on the last day of their trip. Although the spoken dialog corpus was collected in a studio, these observations support the idea that realistic travel experiences were reflected in the experimental tasks, thus validating the task setting and the instructions described in Section 3.

Possibly, the general tasks of type 'Introduction' are part of the fundamental nature of human-human dialog. The ratio of 'Introduction' utterances as shown in Figure 3-11 will most likely decline when human-machine dialog is implemented.



From \ To	Traffic	Introduction	Tourist attraction	Restaurant	Hotel	Facility	Weather	Shopping	Parking	</d>
<d>	37	148	4	2	2	2	1	0	0	0
Traffic	0	19	43	58	27	41	8	1	1	37
Introduction	101	0	28	30	21	8	6	3	0	2
Tourist attraction	24	5	0	36	20	17	3	6	3	65
Restaurant	16	13	47	0	20	18	0	1	1	42
Hotel	20	6	33	10	0	4	0	5	0	19
Facility	22	7	19	15	4	0	2	2	0	20
Weather	8	1	4	2	1	1	0	0	0	3
Shopping	6	0	0	4	1	0	0	0	0	7
Parking	1	0	1	1	1	0	0	0	0	1

(a) Task transition frequency of Task *A*



From \ To	Tourist attraction	Introduction	Restaurant	Traffic	Shopping	Facility	Parking	Weather	</d>
<d>	43	112	16	18	4	0	1	0	0
Tourist attraction	0	8	38	34	15	6	3	1	71
Introduction	72	0	29	19	9	2	1	0	0
Restaurant	29	6	0	12	19	4	1	0	41
Traffic	22	3	18	0	4	2	0	0	47
Shopping	6	2	9	11	0	0	0	0	26
Facility	2	0	1	1	3	0	0	0	7
Parking	2	1	1	1	0	0	0	0	1
Weather	0	0	0	0	0	0	0	0	1

(b) Task transition frequency of Task *B*.

Table 3-1. Task transition frequency of Phase II.

Table 3-1 shows the task transition frequencies of Task A and B. The symbols <d> and </d> indicate the beginning and end of the dialog, respectively. For instance, the first row of Table 3-l(a) indicates that 148 dialogs of Task A started with an introduction. By following the most frequent task transitions in Table 3-1 (a), we obtain the task transition <d> - introduction - traffic - restaurant - tourist attraction - </d>, which suggests that the typical dialog steps of Task A were: 1) explain to the operator about the driving plan; 2) ask for traffic information en route to the destination; 3) ask for restaurant information for lunch; and 4) ask for tourist attraction information after lunch. On the contrary, the path of the most frequent task transitions in Table 3-l(b) is <d> - introduction - tourist attraction - </d>, which does not give us a clear idea of typical dialog steps; this indicates that the driving plan varies according to the speaker. However, this result supports the hypothesis that speakers were most interested in tourist attractions on the second day, as explained above.

The operator sometimes asked the speaker if he/she had any further questions when the dialog halted. In such cases, the information summarized in Table 3-1 could be used to suggest a new task to the speaker (to encourage smooth dialog). For example, the dialog system can suggest the restaurant task and the traffic task after providing guidance on tourist attractions on the second day, e.g. "Where would you like to eat lunch?" and "Do you need traffic information? Traffic congestion is anticipated in the afternoon."

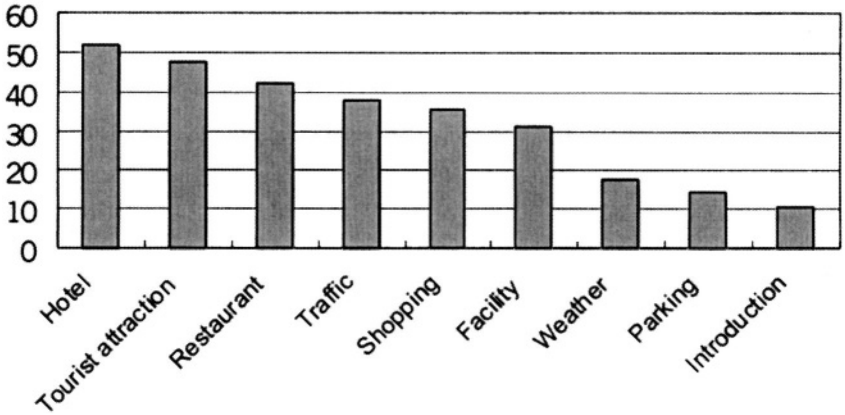


Figure 3-12. Average number of utterances per one task.

Figure 3-12 illustrates the average number of speaker and operator utterances per task. The number suggests the complexity of the task. The greater the number is, the more complex the task is, indicating that the task has many conditions to consider. For example, the hotel information task has the following conditions: 1) the location of the hotel; 2) the style of the hotel (Western or Japanese); 3) the number of persons; 4) the room charge; 5) facilities; 6) tourist attractions near the hotel; etc. The restaurant task has the following conditions: 1) the location of the restaurant; 2) the type of food; 3) price range; 4) acceptability of credit card payments; 5) parking availability; etc. Mixed initiative dialog scenarios must be introduced to handle these tasks in a responsive manner, since speakers do not want to answer question on a one-by-one basis. Narrowing down the conditions to feasible values by considering the context and the driver‘s preferences is also a necessity. For example, the operator might narrow the alternatives to two or three by considering the driving route plan and the locations of the alternatives. The dialog patterns and tactics selected by the operator in the spoken dialog corpus are being examined in order to design responsive HMI dialog scenarios.

6. CONCLUSIONS

A spoken dialog corpus for car telematics services was collected from 137 males and 113 females. Analysis of the spoken dialog corpus revealed that the vocabulary size for speaker utterances was 4,533 words, consisting of 762 proper nouns and 3,771 words other than proper nouns. The average number of dialog tasks per speaker was 8.1. The three most requested types of information in the corpus were traffic information, tourist attraction information and restaurant information. These results are being used to develop and evaluate ASR as well as the dialog scenarios used in the CAMMIA system.

The spoken dialog corpus has several issues which should be addressed in the development of ASR grammars and the dialog scenario for HMIs:

(i) The operator does not talk like a computer.

- The operator uses ambiguous expressions, such as “the route is congested a little bit heavily”.
- The operator does not always state things in a succinct way.

(ii) The speaker does not act like he is talking to a computer

- Many occurrences of filler words such as “eto”, which roughly corresponds to “ummm” in English, are attested in the corpus; humans typically do not inject these filler words into their dialogs with an HMI.

To address these problems, we are planning to use the first prototype dialog system (based on the human-human spoken dialog corpus) to collect human-machine dialogs, which will be used to improve the dialog system in subsequent incremental refinement.

ACKNOWLEDGEMENTS

The authors wish to thank Mr. J. Watanabe of Hitachi Ltd for his valuable comments. The authors are also grateful to Dr. M. Araki, Assistant Professor of Kyoto Institute of Technology, for his ongoing support of the spoken dialog corpus development.

REFERENCES

[1] E. Nyberg, T. Mitamura, P. Placeway, and M. Duggan and N.Hataoka, “DialogXML: Extending VoiceXML for dynamic dialog management,” Proc. of Human Language Technology Conference, 2002

[2] C. Doran, J. Aberdeen, L. Damianos, and L. Hirshman, “Comparing Several Aspects of Human-Computer and Human-Human Dialogues”, Proc. of Second SIGdial Workshop on Discourse and Dialog, Aalborg, Denmark.

[3] C. Cieri, “Resources for Robust Analyses of Natural Language”, Conference ROMAND 2000, Lausanne, France, 2000

[4] NARA INSTIUTE of SCIENCE and TECHNOLOGY, “Morphological Analyzer ChaSen”, http://chasen.aist-nara.ac.jp/

[5] Y. Obuchi, E. Nyberg, T. Mitamura, M. Duggan, S. Judy and N. Hataoka, “ Robust Dialog Management Architecture Using VoiceXML for Car Telematics Systems,” Proc. of Workshop on DSP in Vehicular and Mobile Systems, 2003,“ Proc. of Workshop on DSP in Vehicular and Mobile Systems, Nagoya, Japan, 2003.

[6] M. Tateishi, I. Akahori, S. Judy, Y. Obuchi, T. Mitamura, and E. Nyberg, ”A Spoken Dialog Corpus for Car Telematics Services,” Proc. of Workshop on DSP in Vehicular and Mobile Systems, Nagoya, Japan, 2003.

[7] W3C, “Voice Extensible Markup Language (VoiceXML) Version 2.0 Working Draft,” http://www.w3c.org/TR/voicexml20/

Chapter 4

EXPERIENCES OF MULTI-SPEAKER DIALOGUE SYSTEM FOR VEHICULAR INFORMATION RETRIEVAL

Hsien-Chang Wang[1] and Jhing-Fa Wang[2]

[1]*Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C., Email: wangsj@csie.ncku.edu.tw;* [2]*Department of Electrical Engineering, National Cheng Kung University. Tainan, Taiwan, R.O.C., Email: wangjf@csie.ncku.edu.tw*

Abstract: Currently, most spoken dialogue systems only deal with the interaction between the system and one speaker. In some situations, interactions may occur between several speakers and the system. New functions and improvements need to be made in order to handle a multi-user situation. Studies of the human computer interaction system that involve multiple users are in their initial stages and any papers, lectures or studies on the subject are very limited. For these reasons we are motivated to conduct a further study on multi-speaker dialogue systems. In this chapter, the interactions between the multiple speakers and the system are classified into three types: independent, cooperative, and conflicting interactions. An algorithm for the multi-speaker dialogue management is proposed to determine the interaction type, and to keep the interaction going smoothly. The experimental results show that the proposed algorithm can properly handle the interaction that occurs in a multi-speaker dialogue system, and provides useful vehicular information to the speakers.

Keywords: multi-speaker dialogue, mobile, vehicular information, dialogue management, microphone array

1. INTRODUCTION: FROM SINGLE-SPEAKER TO MULTI-SPEAKER DIALOGUE SYSTEM

It has been several decades since the development and release of the first spoken dialogue system (SDS). Although SDS's can provide convenient human-computer interface (HCI) and many useful functions, most current SDS's address only the interaction between the system and one speaker. In some situations, it is natural and necessary to be able to handle the interaction between multiple speakers and the system. For example, if several passengers in a car are determining where to go for lunch, traditional SDSs would need to be improved in order to deal with the multiple speaker interaction. This motivates the present investigation into the study of multi-speaker dialogue systems (MSDS).

There are many factors to be considered when multiple parties are engaged in an HCI system. Studies of HCI systems that involve multiple users are in their initial stages and any papers, lectures or studies on the subject are very limited. Among the reported studies, Young developed the discourse structure for multi-speaker spoken dialogs based on stochastic model [1]. Bull and Aylett [2] analyzed the timing of turn-taking in dialogues; cross-speaker anaphora was reported by Poesio [3]. This research was based on theoretical studies or the analyses of tagged text-based multi-speaker interactions. Similar papers can be found [4,5,6,7]. Besides these theoretical studies, Matsusaka et al. [8] built a robot that could communicate with multi-users using a multi-modal interface. The robot was equipped with several workstations and cameras to track and process the speaker input. So, in all, previous multi-speaker research [1,2,3,4,5,6,7] either focused on the theoretical discussion of dialogues or required additional expensive heterogeneous hardware for multi-modal input, as reported in [8,9,10,11,12]. The issue that previous research has failed to analyze is the interactions between a dialogue system and speakers. This chapter focuses on the analysis of such interactions and proposes an algorithm for dialogue manager to handle various interactions occurring in an MSDS. Note that two kinds of interaction may occur in a multi-speaker dialogue, as classified below. The first one is the interaction between a speaker and the system (referred to as inter-action), and the other is the interaction between speakers (intra-action). This chapter discusses only the former.

Observation of many multi-speaker interactions lead to the conclusion that during a dialogue, one speaker may either interrupt the utterance of another speaker or wait until the input is finished. That is, the speakers are either making simultaneous input or they utter the input in turn. If an MSDS can handle simultaneous speech inputs, we call it a simultaneous MSDS

(Sim_MSDS); otherwise, it is called a sequential MSDS (Seq_MSDS). In a Seq-MSDS, utterances of speakers are buffered first, and then they are processed together. In this chapter we only consider Seq_MSDS.

In multi-speaker dialogues, speakers may cooperate to accomplish a common goal or negotiate to solve conflicting opinions to achieve the same goal. We defined two types of goals in MSDS (i.e., the individual goal and the global goal). The individual goal is the answer that one speaker wants to know from the inquiry. Since individual goals may conflict with each other, the system should maintain a global goal in which it can integrate the individual goals. The following examples will demonstrate different cases in which individual goals do and do not conflict with each other. Depending on the relationship between two individual goals, the interactions between speakers and the system are classified as one of three types: independent, cooperative, and conflicting. Examples are shown below, where S_1 and S_2 are different speakers:

(i) ***Independent interaction:*** *speakers S_1 and S_2 have independent goals*
S_1: What's the weather in Taipei?
S2: Where is the Tainan train station?

In the first example, the individual goal of each speaker is different and independent.

(ii) ***Cooperative interaction:*** *speakers have a common goal*
S_1: Please find a place to eat.
S_2: I want to eat Japanese noodles.

In second example, the individual goal of S1 is to find a restaurant, and the goal of S_2 is to eat Japanese noodles. The dialogue manager should detect and integrate these individual goals to form the global goal, i.e., a place where Japanese noodles are available.

(iii) ***Conflicting interaction:*** *speakers have conflicting goals*
S_1: Tell me a Chinese restaurant.
S_2: I think we should go to an Italian restaurant.

In third example, S_1 wants to go to a restaurant which supplies Chinese food while; in contrast, S_2 wants to go to an Italian restaurant. Their intentions are similar, but the destinations conflict. The global goal should be adjusted when speaker S_2 has an individual goal different from that of S_1.

In an MSDS, the interactions between the speakers and the system should be handled carefully to keep the dialogue going smoothly. This task is often accomplished by the dialogue manager and is the major issue discussed in this chapter. This chapter is organized as follows: 1) Section 2 describes the major components of an MSDS; 2) Section 3 illustrates the algorithm of a multi-speaker dialogue manager, together with several examples; 3) Section 4 shows the experimental results; finally, the concluding remarks are given in Section 5.

2. FUNDAMENTAL OF MSDS

According to the model provided by Huang et al., [13], a traditional single-speaker SDS can be modeled as a pattern recognition problem. Given a speech input $\boldsymbol{X}$, the objective of the system is to arrive at actions $\boldsymbol{A}$ (including a response message and necessary operations) so that the probability of choosing $\boldsymbol{A}$ is maximized. The optimal solution, i.e., the maximum a posterior (MAP) estimation, can be expressed as following equation:

$$\begin{aligned} A^{*} &= \arg\max_{A} P(A \mid X, S_{n-1}) \\ &\approx \arg\max_{A,S_n} P(A \mid S_n) \sum_{F} P(S_n \mid F, S_{n-1}) P(F \mid X, S_{n-1}) \end{aligned} \tag{1}$$

where $\boldsymbol{F}$ denotes the semantic interpretation of $\boldsymbol{X}$ and $\boldsymbol{S_n}$, the discourse semantics for the $\boldsymbol{n}$th dialogue turn. Note that Eq. (1) shows the model-base decomposition of an SDS. The probabilistic model of an SDS can be found in the work of Young [14, 15].

For the case of multi-speaker dialogue system, assuming that only single-thread speech input is allowed, and speech is input from multiple microphone channels, Eq. (1) can be extended to the formulation below.

$$A^{*} \approx \arg\max_{A} P(A \mid G_n) P(G_n \mid S_n^{1*}, \ldots, S_n^{m*}, G_{n-1}) \tag{2}$$

where $\boldsymbol{G_n}$ denotes the integration of $\boldsymbol{m}$ discourse semantics for the $\boldsymbol{n}$th dialogue turn, it contains all the information in $\boldsymbol{S^i_n}$. And, $\boldsymbol{m}$ is the number of speakers. The discourse semantics $\boldsymbol{S^i_n}$ can be derived using Eq.(3) shown below:

$$S_n^{i*} = \arg\max_{S_n^i} \sum_{F^i} P(S_n^i \mid F^i, S_{n-1}^i) P(F^i \mid X^i, S_{n-1}^i) P(X^i \mid U) \tag{3}$$

where U denotes the multiple input from the multiple microphones and i is the speaker index. Based on Eq. (3), an MSDS can be decomposed into five components as described below:

1. *Active speaker determination:* deciding the active speaker i and their speech input X^i, using model $P(X^i|U)$. In order to aid in the determination of the active speaker along with multiple microphone input, the matched filter can be a useful technique. The output of the matched filter from each microphone is compared with a predetermined threshold to decide the primary channel, i.e., the active speaker. The signals from secondary channels are used to estimate the noise using an adaptive filter. The enhanced signal (i.e., the target speech) is obtained by subtracting the estimated noise from the primary channel signal.

2. *Individual semantic parser:* performing the same parsing process, as in the case of traditional SDS, for each speaker. The semantic model $P(F^i|X^i, S^i_{n-1})$ to parse sentence X^i into semantic objects F^i. This component is often divided into individual target speech recognition and sentence parsing. The speech recognizer translates each speaker's utterance into a word/keyword lattice. Current development of keyword spotters allows them the ability to detect thousands of keywords and yields acceptable results for the applications of SDS. It would be suitable to make use of a keyword spotter in an MSDS in order to detect the meaningful part of a speaker utterance. Our proposed MSDS uses the technique developed by Wu and Chen [16]. Furthermore, we adopt the partial parser which concentrates on describing the structure of the meaningful clauses and sentences that are embedded in the spoken utterance.

3. *Individual discourse analysis:* the discourse model $P(S^i_n|F^i, S^i_{n-1})$ is used to derive new dialogue context S^i_n. This process is also performed for each speaker.

4. *Multiple discourse integration:* the discourses semantics of all speakers are integrated using model $P(G_n|\ S^1_n\ ,\ldots,\ S^m_n\ ,G_{n-1})$. The discourse integration model together with the individual discourse analysis model combines and integrates each speaker's dialogue semantics. The result of discourse integration is sent to the multi-speaker dialogue manager.

5. *Multi-speaker dialogue manager:* to determine the most suitable action by the model $P(A|G_n)$. After multi-speaker speech input is handled properly by these modules, the dialogue manager is responsible for maintaining the

dialogue and keeping it going smoothly. It plays an important role in an MSDS that is described in the next section.

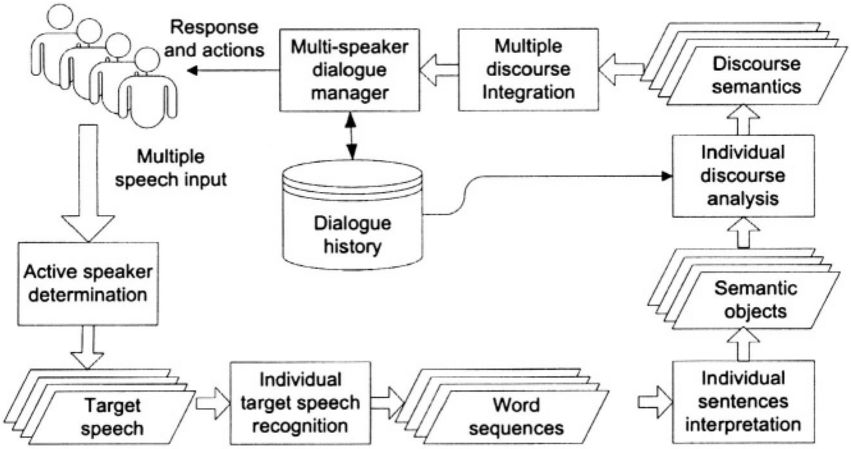


Figure 4-1. Basic components of a multi-speaker dialogue system. The rectangle blocks are the processing units; the parallelograms are multiple outputs derived from the processing units.

3. DIALOGUE MANAGEMENT FOR MSDS

Once the active speaker is determined, the target speech is sent to the speech recognition and natural language processing components. The keyword spotting and partial parsing techniques that are popular in the field of spoken language processing can be adopted in an MSDS. The parsed result will be the most likely words sequence with their part-of-speech tags. They are then fed to the dialogue manager. The dialogue manager maintains the interaction among the system and the multiple speakers and keeps it going smoothly. Fig. *4-1* shows the block diagram of the multi-speaker dialogue management.

In an MSDS, each speaker may have his own individual goal for information retrieval. In contrast to the individual goal, the global goal is the integration of each individual goal. The management of the multi-speaker dialogue has several functions: 1) to interpret intentions and semantics of each individual speaker in order to detect if there is a conflict between speakers, 2) to integrate individual goals into global goals, 3) to determine whether a specific goal is completed, and 4) to generate the response. In this section, we illustrate how the management of MSDS works by giving an algorithm and some examples.

Each time the system receives input from a speaker, natural language processing (as introduced in Sec. 2) is applied to understand the intention and semantics of this speaker. We used a data structure, ***semantic frame SF,*** to record this information, which is defined below (assuming there are *n* speakers):

$$SF_i = (V_D^i, V_{PA}^i, V_{SA_1}^i, V_{SA_2}^i, \ldots),\ i=1\sim n$$

For speaker ***i***, V_D^i represents the domain that speaker mentioned; V_{PA}^i is the primary attribute for this domain, i.e., the purpose of the query; and $V_{SA_k}^i$ is the secondary attributes which specifies additional information needed for this query. Note that the number of secondary attributes varies with domain. Take the inquiry "please show me the route to the train station" as example, the semantic frame will be:

SF = ("NAVIGATION", "DESTINATION", "train station", Null...).

The semantic frame of the current dialogue turn is combined with the previous ones to determine if the goal completed. This determination is based on whether essential information needed for a specific query is enough. For example, if the speaker is querying the weather forecast, the essential information would be the location (ex. city name), the weather type (ex. temperature or rainfall density), and the time (ex. tomorrow or this afternoon). Once a goal is completed, the system may perform database queries and generate a proper response to the speaker. If some essential information is missing or the speaker interactions are in conflict, further confirmation and a repair processes should be undertaken to realize the final intention of the speakers.

In examples 1,2, and 3, we illustrate the cases of 1) speakers who have independent individual goals, which can be solved easily; 2) speakers with conflicting individual goals, in which the system must resolve this problem before further information can be relayed to the speakers; and 3) speakers who have a common goal, which requires that they provide the necessary information for the system in a mutually cooperative fashion.

Input: Partial parsing results of speech recognition for each speaker, denoted as PP_1, PP_2, ... , PP_m, where *m* is the number of total speakers.

Output: response to speakers.

Step 1: Initialization

- Initialize the semantic frames SF_i to be NULL.

$SF_i = (V_D^i, V_{PA}^i, V_{SA1}^i, V_{SA2}^i, \ldots)$, $i=1\sim m$

For speaker *i*, V_D^i represents the mentioned speaker domain; V_{PA}^i is the primary domain attribute; and V_{SAj}^i are secondary attributes, where *j* varies with the domain.

- Initialize the dialogue history lists, *Hi,* for each speaker to NULL.

Step 2: Determine the semantic frame

Apply NLP techniques to PP_i to determine the corresponding semantic frame SF_i. Semantic frame *SFi* for this turn is copied to the history *Hi*.

Step 3: Determine interaction type for any speaker pair *(i, j)*

```
If $V_D^i = V_D^j$
        if $V_{PA}^i \neq V_{PA}^j$ then **Cooperative interaction**
        else_if $V_{PA}^i = V_{PA}^j$ then **Conflicting interaction**
else_if $V_D^i \neq V_D^j$
                then **Independent interaction**
```

Step 4: Semantic integration

SF_i*'s* and H_i*'s* are integrated to determine if a goal is completed.

(Detailed method for semantic integration is listed in Section 3.1.)

Step 5: Determine the accomplished goal(s)

For each speaker, check if the necessary information slots for completing a goal are filled or not. A goal is completed ⇔ (V_D Null) AND (V_{PA} Null) AND (V_{SA} Null)

Step 6: Decision

If any goal is completed, go to Step 7; else go to Step 8.

Note that the conditions for a goal to be completed are definable for each domain.

Step 7: Response: Perform database query and generate response to the user according to the goal(s) found in Step 3. Go to Step 8.

Step 8: Iteration: Accept next input and go to Step 2.

Figure 4-2. Algorithm 1: The algorithm for multi-speaker dialogue management

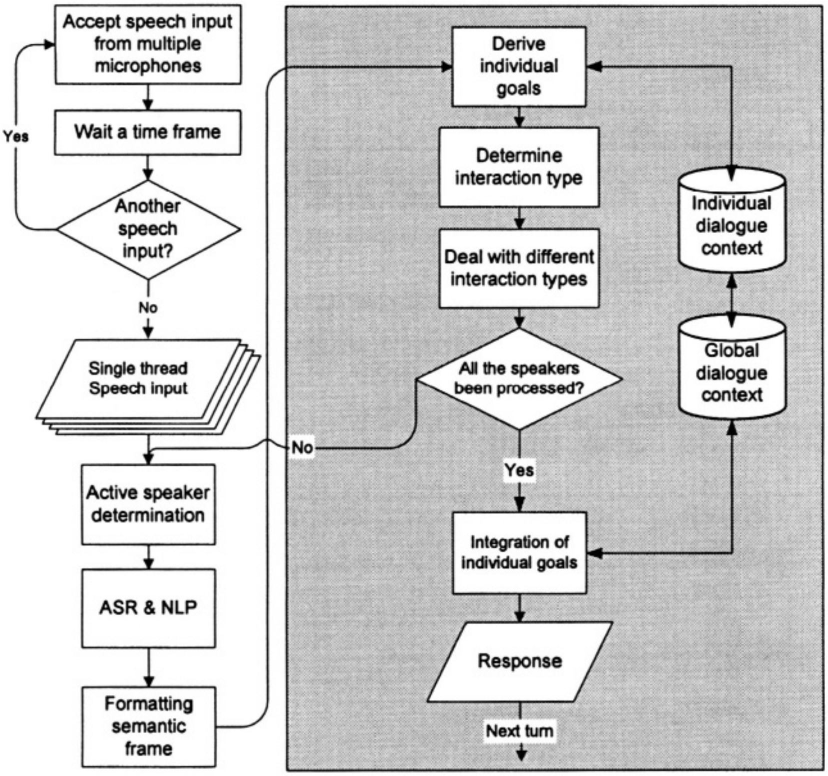

Accept speech input from multiple microphones
Wait a time frame
Another speech input?
Yes
No
Single thread Speech input
Active speaker determination
ASR & NLP
Formatting semantic frame
Derive individual goals
Determine interaction type
Deal with different interaction types
All the speakers been processed?
No
Yes
Integration of individual goals
Response
Next turn
Individual dialogue context
Global dialogue context

Figure 4-3. Block diagram of the multi-speaker dialogue manager

Example 1. Speakers have different individual goals.

Time index	Action	Content
1	(Speaker$_1$) inputs	"I want to go to the city hall".
2	(Speaker$_2$) inputs	"Tell me the weather in Taipei"

3	(System) derives *SFi*	SF$_1$=("NAVIGATION", "ROUTE", "city hall", Null...)
		SF$_2$=("WEATHER", "LOCATION", "Taipei", Null...)

4	(System) checks goal completeness	Speaker$_1$=TRUE
		Speaker$_2$=TRUE

5	(System) checks if conflict happened	NO
6	(System) generates response	"The weather in Taipei is rainy." "The city hall is about 450 meters away, please follow the instructions."

Example 2. Speakers have conflicting individual goals.

Time index	Action	Content
1	(Speaker$_1$) inputs	"Find me a Chinese food restaurant".
2	(Speaker$_2$) inputs	"No, I want to eat Italian food"

3	(System) derives *SFi*	SF$_1$=("NAVIGATION", "ROUTE", "destination=restaurant", "attribute=Chinese food", Null…)
		SF$_2$=("NAVIGATION", "ROUTE", "destination=restaurant", "attribute=Italian food", Null…)

4	(System) checks goal completeness	Speaker$_1$=TRUE
		Speaker$_2$=TRUE

5	(System) checks if conflict	YES
6	(System) resolves conflict	"Please specify again, do you want Chinese food or Italian food"

Example 3. Speakers have a common goal.

Time index	Action	Content
1	(Speaker$_1$) input	"I want to know the route to …".
2	(System) derives SFi	SF$_1$=("NAVIGATION", ”ROUTE", Null…)
3	(System) checks goal completeness	Speaker$_1$=FALSE, (no DESTINATION)
4	(System) generate response	"Please specify the destination."
5	(Speaker$_1$) input	"To the nearest gas station"
6	(Speaker$_2$) input	"And, how far is the gas station?"

7	(System) combines new SFi with old ones	SF$_1$=("NAVIGATION", ”ROUTE", "destination=gas station", "attribute=nearest" Null…)
		SF$_2$=("NAVIGATION", "DISTANCE", "destination=gas station", "attribute=nearest", Null…)

8	(System) checks goal completeness	Speaker$_1$=YES
		Speaker$_2$=YES

9	(System) checks if conflict	NO
10	(System) generates response	"The nearest gas station is 540 meters ahead, please continue straight ahead."

These examples demonstrate three types of interaction between two speakers. For the cases in which more than two speakers are involved, the same approach is applied to check the interaction type and goal completeness, and generate the response to the speakers.

4. EXPERIMENTAL RESULTS

To test whether the proposed methods were feasible, an experimental environment was set up in a 1,600 CC automobile. The recording device was a notebook computer together with a PCMCIA multi-channel recording card and four omni-directional microphones. We developed an MSDS which was capable of answering user queries in three application domains, i.e., route guidance, weather forecasting, and stock prices. A GPS (global positioning system) receiver was mounted on the car to acquire its current position. The information about weather forecasts and stock prices was stored in a remote server that was able to get up-to-date information through the Internet. When inquires about these domains were made, a short message was issued from an embedded cell phone to the server. After a database query on the server-end, the query result, in short message format, was sent back to the cell phone and was interpreted, resulting in the desired information.

Thirty-two speakers aged 17 to 35 participated in our experiment. Before the experiment, testers were briefly informed regarding the capability and domains of the system. Two types of experiment were carried out in our work. The first was for active speaker determination because knowing the active speaker is essential in order for the dialogue manager to make a correct response. The second was the evaluation of the proposed MSDS; both subjective and objective evaluation metrics are reported in our experiments.

4.1 Experimental Results of Active Speaker Determination

For the experiment of active speaker determination, we set up four different configurations of microphone placement, as shown in Figure 4-4. Four speakers, S_1, S_2, S_3, and S_4, were in the upper-left, upper-right, lower-left, and lower-right corner of the car, respectively, as indicated in Configuration 1 of Figure 4-4.

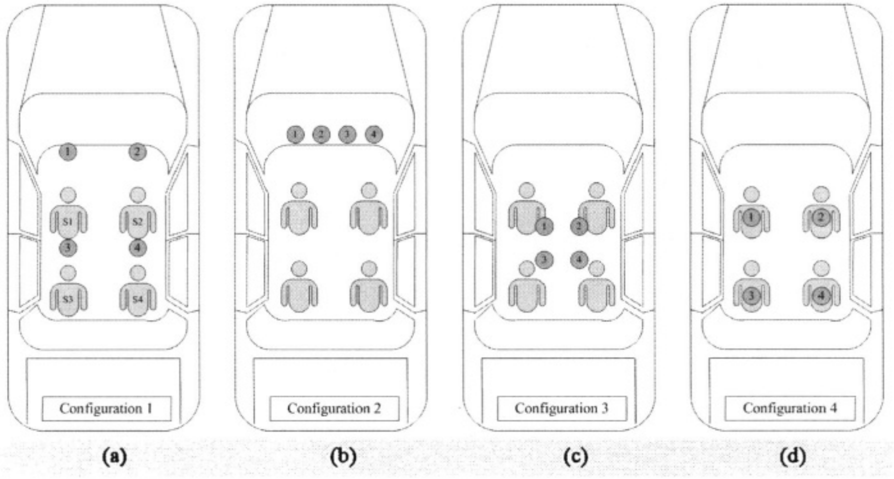


Figure 4-4. Different configurations of microphone placement in the car environment: (a) A microphone was placed in front of each speaker in the car; (b) Four microphones, linear formation, were placed above the windshield of the car. The distance between two microphones was 20 cm; (c) Four microphones, in square formation, were placed in the center of the car ceiling. The side length of the square is 20 cm; (d) The microphones were attached on the seat belts of each passenger.

We assessed the rate of correct active speaker determination. Each tester uttered thirty short words for testing. Three conditions of car speed were maintained in this experiment, i.e., idle, in-city, and on-highway conditions. The idle condition means that the car was ignited but remained stationary. For the in-city and on-highway conditions, respectively, the car speed was kept at 0~50 km/h and 70~100 km/h. Table 4-1 shows the speech identification results of our experiments.

As shown in Table 4-1, the 4th configuration achieved the best performance of ASD, since the vibration caused by the car's engine was absorbed by the soft seatbelt. Configuration 3 was the worst because the threshold was hard to determine (the difference between speakers was not noticeable). For speakers in the front seats, configurations 1, 2 and 4 yielded similar results. The reason was that under these configurations, the signals from primary and secondary channels contained noticeable differences.

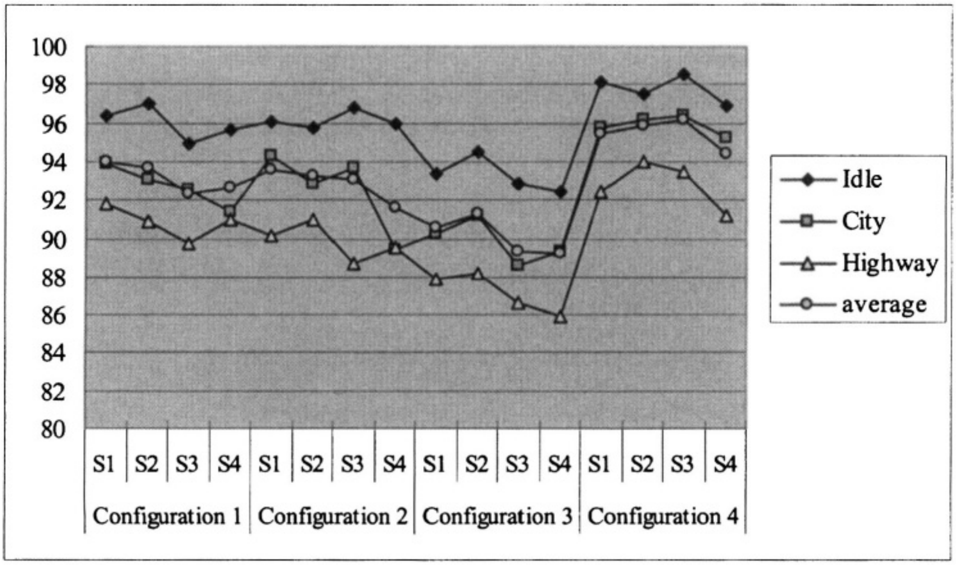


Figure 4-5. Correct rate of the active speaker determination. The experimental setup was: four speakers S1, S2, S3, and S4 sat in different corner of the car; microphones were placed in the car in four configurations; and the speeds of car were maintained for three conditions (idle, city, and highway).

4.2 Statistics of the interactions in an MSDS

In order to determine the interactive behavior in a multiple speaker dialogue, the thirty-two testers were divided further into eight groups for the second experiment. Each group of testers performed multi-speaker dialogues six times. Each tester was given the freedom of deciding whether he wanted to join in the dialogues. That is, during a multi-speaker dialogue, the number of parties could be one, two, three, or four. Also, the testers were free to choose one of the three task domains, i.e., navigation guide, weather information, or stock price information. Within this paradigm, we computed three statistics about the interactions of an MSDS. The first was how many parties engaged in each MSDS. The second was the percentage of interaction types that occurred in the MSDS. The last was the rate of correct interaction type determination.

# of parties	# of dialogues performed	Percentage of the interaction types			
		Independent	Cooperative	Conflict	Total
1	5	10 %	NULL	NULL	10 %
2	19	15 %	17 %	8 %	40 %
3	15	10 %	8 %	13 %	31 %
4	9	7 %	7 %	5 %	19 %
Total	48	42%	32%	26%	100%

Table 4-1. **Statistics of the interactions in the experiments.**

The results of the first two experiments are shown in Table 4-1. As shown in Table 4-1, it would be natural for multiple persons to interact together in order to derive their desired information. Only five dialogues (10%) are single-speaker dialogues. It can also be observed from Table 4-1 that the three interaction types happened with almost the same probability. The statistics in Table 4-1 suggest that the study of MSDS is a necessary one.

As mentioned in Section 3, the determination of interaction type involved comparing the domain (V_D), primary feature (V_{PA}), and secondary features (V_{SA}) for speakers. The correct rates of interaction type determination were 98.3%, 95.7% and 94.1% for independent, cooperative, and conflict interactions, respectively. The wrong determinations occurred in cases in which the speaker omitted the domain slot and provided just primary or secondary feature slot information. The system "guessed" the domain slot based on the identification of the other slots, which may have resulted in incorrect determinations of the interaction type.

4.3 Experimental Results for the MSDS Evaluation

The evaluation of a spoken dialogue system can be classified as *objective* and *subjective* as indicated by Danieli and Gerbino [17], Hirschman and Pao [18], and Walker et al. [19]. Objective metrics can be calculated without human judgment, and in many cases can be logged by the SDS so that they can be calculated automatically. Subjective metrics require subjects using the system, and/or human evaluators to categorize the dialogue or utterances with various qualitative measures. Both subjective and objective evaluations were used in the experiment. The metrics were:

1. percentage of different interaction types (i.e., independent, cooperative, conflict)

2. percentage of completed tasks
3. average number of turns
4. task completion time
5. mean system response time
6. mean length of utterances
7. percentage of correct answers
8. user satisfaction
9. willingness to use the system

Metrics 1 to 7 are the objective evaluations while the others are subjective. For the evaluation of the MSDS, we adopt the 4th microphone configuration (i.e., microphones are embedded on the seatbelts of each speaker). The testers were divided into three groups, namely cooperative, independent, and conflicting groups of speakers. Before the experiment, the scenario that describes the interaction of these three types is given to the testers. The experimental results are shown in Table 4-2.

As shown in Table 4-2, the task completion times seem a little lengthy. This is caused by the SMS (short message system) communication between the embedded cell phone and the server. More than half of the testers were satisfied with the system's ability and performance, and about half of the testers were willing to use this system if it became commercially available.

Metrics	Independent	Cooperative	Conflicting	Average
1. Interaction types (%)	42	32	26	33.33
2. Task completion rate (%)	80.2	76.3	72.5	76.33
3. Average number of turns	6.2	8.5	10.1	8.27
4. Task completion time (secs)	37.1	48.2	45.7	43.67
5. Mean system response time (secs)	1.9	2.1	2.4	2.13
6. Mean length of utterances (words)	9.1	7.5	6.3	7.63
7. Correct answers (%)	83.6	71.3	80.1	78.33
8. User satisfaction (0~10)	6.03	6.62	5.92	6.19
9. Willingness to use system again (%)	63.1	70.5	52.7	62.1

Table 4-2. Experimental results of different evaluation metrics for the MSDS system under different interaction types

5. CONCLUSIONS AND FUTURE WORK

In this chapter, we have addressed important issues for the development of a multi-speaker dialogue system. The interaction types between the speakers and the system are analyzed, and, an algorithm of the multi-speaker dialogue management is presented. Based on the proposed techniques, an MSDS system was built to provide vehicular navigation information and assistance in the car environment where every passenger may want to interact with the system. The proposed MSDS system can interact with multiple speakers and resolve conflicting opinions. Speakers are also able to acquire multi-domain information independently or cooperatively.

Since our research is in the initial stage, only interaction (c.f. intra-action) between speakers is studied in this manuscript. To model both the interaction and intra-action in an MSDS is a more difficult task and requires further studies, both in the theoretical as well as in the practical arena. Research concerning multi-speaker spoken dialogue systems (MSDS) is in its initial stage and we hope that our works will help to encourage further research into the techniques of MSDS.

As future works, we plan to investigate the communication model for both inter-action and intra-action in an MSDS. We will try to combine blind source separation (BSS) techniques to deal with simultaneous MSDS, i.e., to allow speakers to utter simultaneously in order to provide a more natural and convenient MSDS system.

REFERENCES

[1] Young, S.J., “Talking to Machines (Statistically Speaking)”, in the Proceeding of International Conference on Spoken Language Processing, Denver, Colorado, 2002.

[2] Bull, M. and Aylett, M., “An Analysis of The Timing of Turn-Taking in A Corpus of Goal-Oriented Dialogue”, in Proceedings of the International Conference on Spoken Language Processing, volume 4, pages 1175-1178, Sydney, Australia, 1998.

[3] Poesio, M., “Cross-speaker Anaphora and Dialogue Acts”, in Proceeding of the workshop on Mutual Knowledge, Common Ground and Public Information ESSLLI Summer School, 1998.

[4] Berg, J. and Francez, N., “A Multi-Agent Extension of DRT, .Technical report of Laboratory for Computation Linguistics”, in Proceeding of the 1st International Workshop on Computational Semantics, pp. 81-90. University of Tilburg, 1994.

[5] Cohen, P.R., Coulston, R. and Krout, K., “Multiparty Multimodal Interaction: A Preliminary Analysis”, in Proceeding of International Conference on Spoken Language Processing, 2002.

[6] Hinkelman, E.A. and Spaceman, S.K., “Communication with Multiple Agents”, in Proceedings of the 15th International Conference on Computational Linguistics (COLING’94), vol. 2, pp. 1191-1197, Kyoto, Japan, 1994.

[7] Shankar, T.R., VanKleek, M., Vicente, A. and Smith, B.K., "Fugue: A Computer Mediated Conversational System that Supports Turn Negotiation", in 33rd Hawaii International Conference on System Sciences, Los Alamitos: IEEE Press, 2002.

[8] Matsusaka, Y., Tojo, T., Kubota, S., Furukawa, K., Tamiya, D., Hayata, K., Nakano, Y. and Kobayashi T., "Multi-person Conversation via Multi-modal Interface – A Robot who Communicate with Multi-user", in Proceeding of EuroSpeech'99, pp. 1723-1726, 1999.

[9] Johnston, M., Bangalore, S., Stent, A. Vasireddy, G. and Ehlen, P. (2002). "Multimodal Language Processing for Mobile Information Access", in Proceeding of International Conference on Spoken Language Processing, 2002.

[10] Marsic, I., "Natural Communication with Information Systems", Proceedings of the IEEE, Vol. 88, pp. 1354-1366, 2002.

[11] Rössler, H., Wajda, J.S., Hoffmann, J. and Kostrzewa, M., "Multimodal Interaction for Mobile Environments", in Proceeding of International Workshop on Information Presentation and Natural Multimodal Dialogue, 2001.

[12] Traum, D. and Rickel, J., "Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds", in Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2. pp.766-773, 2001.

[13] Huang, X., Acero, A. and Hon, H.W., Spoken Language Processing, New Jersey: Prentice Hall, 2001.

[14] Young, S.R., "Discourse Structure for Multi-speaker Spontaneous Spoken Dialogs: Incorporating Heuristics into Stochastic RTNS", in Proceeding of International Conference on Acoustic and Speech Signal Processing, pp. 177-180, 1995.

[15] Young, S.J., "Probabilistic Methods in Spoken Dialogue Systems", Philosophical Transactions of the Royal Society (Series A) 358(1769): pp.1389-1402, 2000.

[16] Wu, C.H. and Chen, Y.J., "Multi-Keyword Spotting of Telephone Speech Using a Fuzzy Search Algorithm and Keyword-Driven Two-Level CBSM," Speech Communication, Vol.33, pp.197-212, 2001.

[17] Danieli, M. and Gerbino, E., "Metrics for evaluating dialogue strategies in a spoken language system". in Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, Stanford, CA, pp. 34–39, 1995.

[18] Hirschman, L. and Pao, C., "The cost of errors in a spoken language system", in Proceedings of the Third European Conference on Speech Communication and Technology, Berlin, Germany, pp. 1419–1422, 1993.

[19] Walker, M.A., Litman D.J., Kamn C.A., and Abella A., "Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies", Computer Speech and Language, vol. 12, pp. 317-347, 1998.

This page intentionally left blank

Chapter 5

ROBUST DIALOG MANAGEMENT ARCHITECTURE USING VOICEXML FOR CAR TELEMATICS SYSTEMS

Yasunari Obuchi[1], Eric Nyberg[2], Teruko Mitamura[2], Scott Judy[2], Michael Duggan[3], Nobuo Hataoka[4]
[1]*Advanced Research Laboratory, Hitachi Ltd., Kokubunji, Tokyo 185-8601, Japan,;*
[2]*Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA;*
[3]*Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA;*
[4]*Central Research Laboratory, Hitachi Ltd., Kokubunji, Tokyo 185-8601, Japan*
Email: obuchi@rd.hitachi.co.jp

Abstract: This chapter describes a dialog management architecture for car telematics systems. The system supports spontaneous user utterances and variable communication conditions between the in-car client and the remote server. The communication is based on VoiceXML over HTTP, and the design of the server-side application is based on DialogXML and ScenarioXML, which are layered extensions of VoiceXML. These extensions provide support for state-and-transition dialog programming, access to dynamic external databases, and sharing of commonly-used dialogs via templates. The client system includes a set of small grammars and lexicons for various tasks; only relevant grammars and lexicons are activated under the control of the dialog manager. The server-side applications are integrated via an abstract interface, and the client system may include compact versions of the same applications. The VoiceXML interpreter can switch between applications on both sides intelligently. This helps to reduce bandwidth utilization, and allows the system to continue even if the communication channel is lost.

Keywords: VoiceXML, Dialog Management, Telematics, Speech Recognition.

1. INTRODUCTION

Spoken dialog management in car telematics is a challenging topic for speech and language technology research. The various challenges include efficient creation of dialog scenarios, accurate analysis of the user's utterances, and management of the communication between the client and the server. Because of strict limitations on computational resources and communication bandwidth, the client and the server systems need to divide their tasks in an appropriate manner. VoiceXML[l] provides a useful basis for the design of a system architecture where the server system provides the minimal information necessary to guide the dialog, and the client system transmits the minimal information necessary to describe the user's input.

Carpenter, et al.[2] have proposed a framework for server-side dialog management; since VoiceXML does not directly support the modeling of dialogs as state-transition networks, their framework assumes that the dialog manager[3] controls the entire flow of the dialog, and sends small segments of VoiceXML (representing single dialog turns) to the client. However, for mobile applications such as car telematics systems, the communication channel is narrow and unstable, and we therefore prefer to send the client a single VoiceXML document that includes several dialog turns. In previous work, we have proposed two extensions to VoiceXML: DialogXML and ScenarioXML[4]. DialogXML supports a higher-level model of dialog flow using states and transitions; ScenarioXML provides a systematic mechanism for smooth transition between multiple active dialogs, along with access to external databases and information sources. These extensions are essential for dialog management in car telematics systems. The ScenarioXML dialogs written by the developer are compiled into VoiceXML documents that can be interpreted by the client. On the server side, following the work reported in [2] and [4], Java Server Pages (JSP)[5] are used by the dialog manager to create VoiceXML documents dynamically, so that a particular application may also incorporate information from external databases which is accessed in real time.

Another challenge for an in-vehicle dialog system is accurate analysis of the user's utterances. In a system that has rich computational and/or communication resources, such as a telephony gateway, a large-vocabulary continuous speech recognition (LVCSR) system (e.g., SPHINX[6]) and a large scale natural language processing (NLP) system (e.g., KANTOO[7]) can be integrated. However, in a client system with limited resources, the complexity of the analysis algorithms must be simplified. Our system uses a simple speech recognizer with a regular grammar, and a set of small grammars and lexicons for NLP processing. A (grammar, lexicon) pair defines a task, and the dialog manager can activate one or more tasks by enumerating a specific set of (grammar, lexicon) pairs. Using this approach, a developer can develop a robust dialog system for a particular task in a straightforward manner. Such grammars and lexicons may be created by hand, or derived from corpora that exemplify typical sentences from the specified task.

The third challenge addressed by our system is task switching between the client and the server systems. Sometimes the in-vehicle client loses its connection to the remote server, and must continue to interact with the user as a stand-alone system. In such situations, the client system continues the dialog in a reduced way, providing limited information to the user. After the connection is re-established, the client and the server negotiate to synchronize the current dialog context, and start or continue the next task as appropriate.

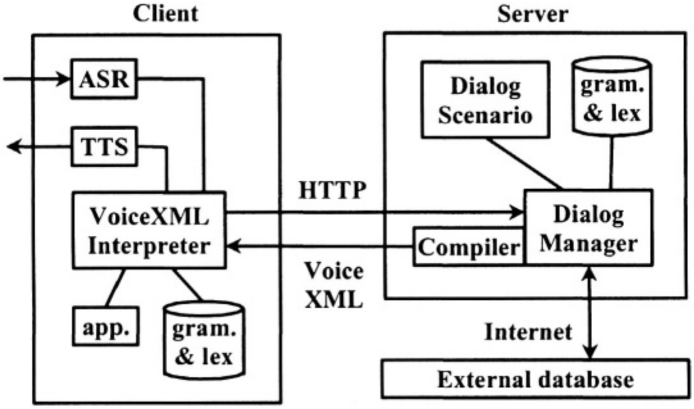


Figure 5-1. System architecture.

2. SYSTEM ARCHITECTURE

Figure 5-1 illustrates the architecture of the car telematics system described in this chapter. In the client system, the VoiceXML Interpreter interacts with the user via automatic speech recognition (ASR) and text-to-speech (TTS) interfaces. We use the VoiceXML Interpreter developed by Hitachi CRL[8]; it supports most of the functions defined in VoiceXML 2.0[1], and also includes an additional input/output channel to allow asynchronous communication with an internal application such as a GPS navigator. Therefore, although the system usually communicates by sending

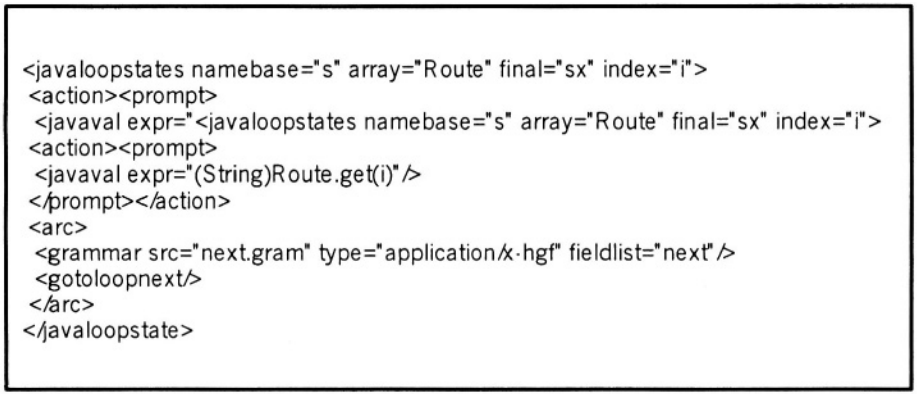
HTTP requests to the Dialog Manager (DM) and receiving VoiceXML documents in return, it can also work in standalone mode by utilizing internal applications. For example, an asynchronous signal from a GPS module can interrupt an ongoing dialog (e.g., a query about restaurant information) when the vehicle approaches an intersection where a turn is required. Grammars and lexicons are stored in the client and the server, and the DM specifies the location of the grammar and the lexicon to use, simply by including a specific URI in the VoiceXML document. If a server-side grammar and lexicon are required for the task, they are sent with the VoiceXML document. On the server side, the DM provides centralized coordination and controls the dialog according to pre-defined dialog scenarios. In order to create a VoiceXML dialog that includes dynamic data (such as specific navigation directions), the DM communicates with external databases through the Internet. External databases provide various types of static and dynamic information, such as traffic and parking conditions, nearby restaurant names, and the current weather forecast. The VoiceXML compiler is a part of the DM and it compiles the ScenarioXML into VoiceXML format for the VoiceXML Interpreter. Grammars and lexicons are stored on the server side, and the DM transmits them along with the VoiceXML output if the client side system does not already contain them.

3. EXTENSIONS OF VOICEXML

Layered extensions of VoiceXML have been proposed as a way to realize straightforward development of VoiceXML applications with dynamic content [4]. The DialogXML layer was proposed to enable the developer to write any dialog flow using a state-and-transition model. Although it is possible to write an equivalent form-filling and if-then style VoiceXML document by hand, state-and-transition style dialog creation is more efficient for developers, especially as dialogs grow beyond simple menu selection to more complex dialogs. The DialogXML compiler translates DialogXML dialogs into VoiceXML, which is generally much longer (by at least a factor of two or three) and more difficult to read than the original DialogXML. The ScenarioXML layer was proposed as a way to specify transitions between active dialogs, and to support dynamic content retrieved from external databases and other information sources. Since most information is dynamic in real-life applications, it is necessary to generate the DialogXML output with up-to-date information included at run-time. In our system, JSP technologies are used for that purpose. The higher-level control of the JSP engine is useful when writing dialog scenarios, since a dialog template can be integrated with Java calls that insert dynamic content. A ScenarioXML compiler was developed to translate ScenarioXML documents into DialogXML [4].

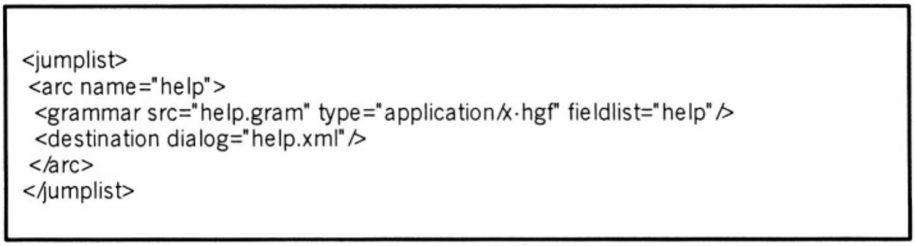
Figure 5-2 shows an example of ScenarioXML; a loop of similar states is described in a higher level programming style, and a Java function with an incrementing argument is called to access the external database. In this example, each function call gets the next instruction for route guidance. After providing the instruction, execution moves to the next state and gets another instruction. Figure 5-3 shows another example of ScenarioXML. Since there are some typical patterns that could be used anywhere in the dialog, those patterns are described as *common arcs.* In this example, the Help dialog can be accessed from any other active dialog if the appropriate common arc is inserted.

Fig. 5-4 shows a segment of DialogXML generated from the examples of Fig. 5-2 and Fig. 5-3 by invoking the ScenarioXML compiler. Each state has an action and a set of arcs. In this example, the action and the first arc were generated from the main ScenarioXML shown in Fig. 5-2, and the second arc was added as a common arc from Fig. 5-3. Since route guidance tasks consist of several steps (directions from an origin to the destination), they will include several states that are similar to this example. Finally, the DialogXML document is compiled again to generate the VoiceXML that can be interpreted by the VoiceXML Interpreter.



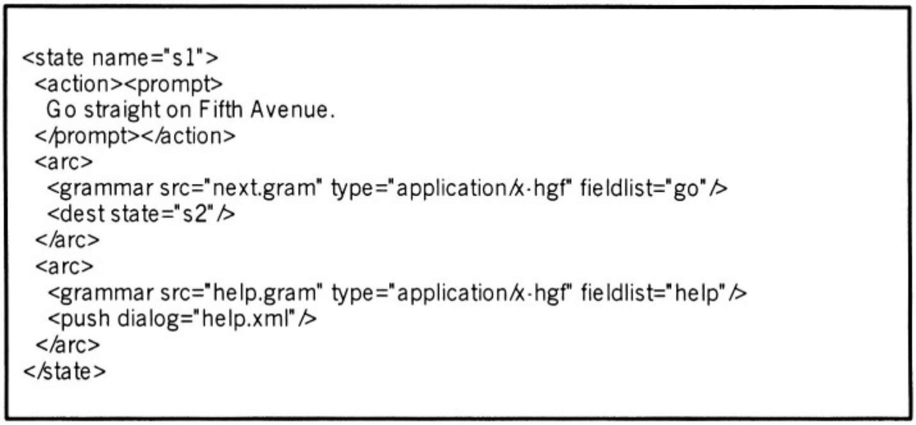
```
<javaloopstates namebase="s" array="Route" final="sx" index="i">
 <action><prompt>
  <javaval expr="<javaloopstates namebase="s" array="Route" final="sx" index="i">
 <action><prompt>
  <javaval expr="(String)Route.get(i)"/>
 </prompt></action>
 <arc>
  <grammar src="next.gram" type="application/x-hgf" fieldlist="next"/>
  <gotoloopnext/>
 </arc>
</javaloopstate>
```

Figure 5-2. Example of ScenarioXML: Loop and access to external database.



```
<jumplist>
 <arc name="help">
  <grammar src="help.gram" type="application/x-hgf" fieldlist="help"/>
  <destination dialog="help.xml"/>
 </arc>
</jumplist>
```

Figure 5-3. Example of ScenarioXML: common arc.



```
<state name="s1">
 <action><prompt>
  Go straight on Fifth Avenue.
 </prompt></action>
 <arc>
  <grammar src="next.gram" type="application/x-hgf" fieldlist="go"/>
  <dest state="s2"/>
 </arc>
 <arc>
  <grammar src="help.gram" type="application/x-hgf" fieldlist="help"/>
  <push dialog="help.xml"/>
 </arc>
</state>
```

Figure 5-4. Example of DialogXML.

A more complicated example is shown in Fig. 5-5. There are two flows of the main dialog and two types of common arcs. In this figure, every arc is related to a specific grammar. It means that the control of the dialog flow is tightly related with grammar selection. This principle is described in detail in the following section.

4. GRAMMARS AND LEXICONS

In VoiceXML, a *grammar* specifies (for a particular dialog state) the set of allowed words and the structure(s) of allowable sentences using those words (defined in terms of legal part-of-speech sequences). To avoid confusion, we refer to the former aspect of VoiceXML grammars as a “lexicon”, and the latter aspect as a “grammar”; the pairing of a lexicon and a grammar in VoiceXML is referred to as a “<grammar>.” A <grammar> operates on an input to capture a set of attribute-value pairs from the user’s utterance. One can claim that the widest coverage of the user's utterance could be achieved by using a LVCSR module and a statistical language model. In such a case, an NLP module must be used to extract information about a specific attribute-value pair. However, it is not reasonable to implement such modules in the client system because the computational resources are limited in the vehicle. Therefore, we use a simple speech recognizer with a regular grammar and a small lexicon, defined as a <grammar>, in the ASR part of the client system.

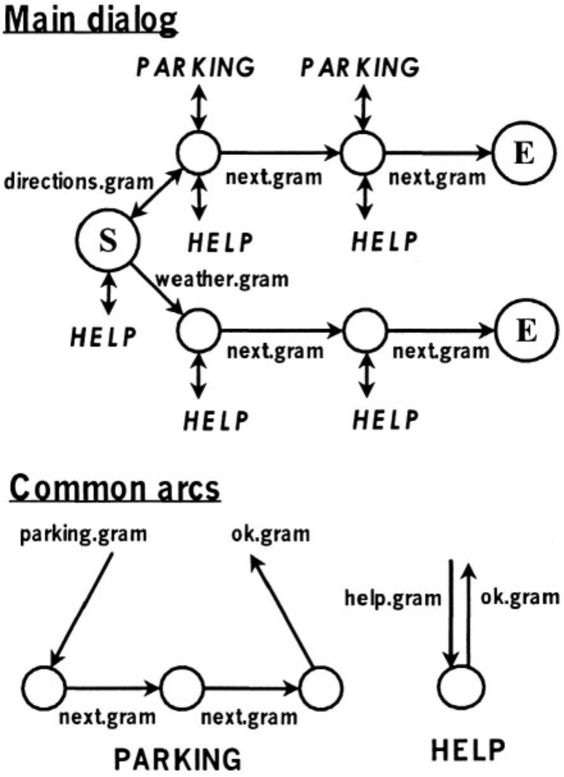


Figure 5-5. Transitions and grammars.

Another important role of a <grammar> is control of the dialog flow. Fig. 5-5 shows that each arc includes a <grammar>, specified by a filename with the ".gram" extension. Since the VoiceXML specification allows us to include multiple <grammar>s in a single form, we can easily split the flow of the dialog by checking which <grammar> covers the user's utterance. If we have the table of <grammar> names and the table of attribute-value pairs allowed in each <grammar>, the developer can write ScenarioXML documents simply by referring to the appropriate <grammar>s.

In speech systems, it is important to keep <grammar>s small; as perplexity increases, the likelihood of recognition errors will also increase. Therefore, building <grammar>s requires a balance between two competing constraints: minimizing the <grammar> size for optimal recognition accuracy, and expanding the grammar to achieve sufficient coverage for the given task. A skilled programmer may be able to construct such <grammar>s by hand, but it would be useful to have a system that can create such <grammar>s automatically.

Figure 5-6 describes a procedure for automatic <grammar> creation using a corpus[9]. We refer to this process as “grammar compilation”; a basic grammar is written by hand and then compiled into a form that is harder for humans to read, but more suitable for the specific task. First we create a unification grammar (UG)[10] that is written in a human-readable format that is familiar to computational linguists. The UG is then compiled into a context-free grammar (CFG) by expanding all constraints. For a single UG rule, a set of CFG rules is created where each CFG rule corresponds to a single set of legal feature-value assignments on the right-hand side of the original UG rule. Then the CFG is compiled to a regular grammar (RG) by introducing an upper limit of the number of recursions allowed for recursive rules[11]. The derived RG can be expressed as a finite state machine (FSM), as shown in Figure 5-6. Then the FSM is used to parse the sentences in the corpus. After parsing all sentences, only the nodes and arcs in the FSM that were activated by at least one sentence are retained, and other nodes and arcs are deleted. This procedure creates a reduced regular grammar that covers all sentences in the corpus and is smaller than the original grammar.

On the other hand, we have yet to create an automatic procedure for lexicon compilation. If we use only the words from the original corpus that were recognized by arcs in the grammar, the reduced grammar’s coverage will be very weak. The utterance “How can I get to Tokyo?” would not be covered, even if the corpus includes the utterance “How can I get to Kyoto?”. However, if we generalize arcs to recognize any words matching the appropriate part of speech, the degree of generalization would be too strong, resulting in poorer speech recognition performance. To boost grammar coverage, we currently utilize semantic word recognition categories (e.g., LOCATION) which are created for each dialog task. Automatic lexicon compilation using a corpus is part of our ongoing research.

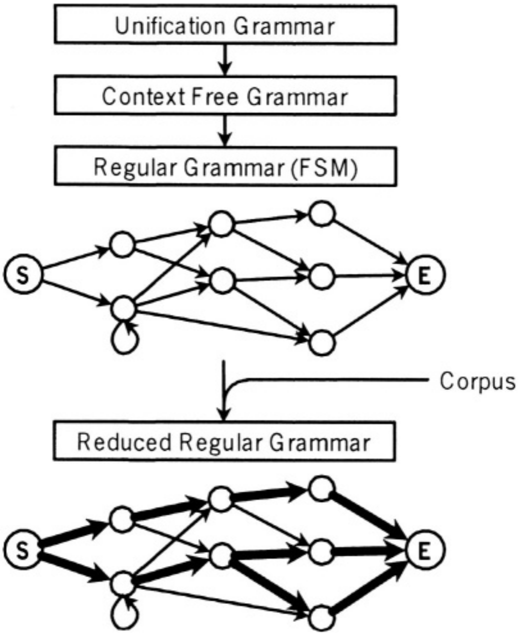


Figure 5-6. Grammar compilation using corpus.

5. SWITCHING EXTERNAL/INTERNAL APPLICATIONS

In car telematics systems, communication between the server and the client can be unstable. The system must exhibit robust behavior when a communication channel is suddenly disconnected without warning. The VoiceXML specification includes an “error.badfetch” event, which signals that an error occurred when fetching a requested document. Our VoiceXML documents therefore include event handlers for “error.badfetch” that switch dialog control to a local, compact dialog management application residing on the client side. For example, if the dialog is about traffic guidance, the internal application may know the route from the current position to the desired destination, but it will not have access to dynamic (real-time) information such as current traffic conditions. If the user asks about traffic conditions in the absence of an established communication channel, the system would reply “I’m sorry. Currently I can’t access that information.” The local dialog manager will enter a wait state, and poll the remote server periodically in an attempt to re-establish the communication channel until it is forced to proceed to the next dialog task by the user's command.

It is also possible to store the complete DM application on the client side if the application does not require any dynamic information. For example, voice control of the vehicle air conditioner can be achieved on the client side with no need for server-side dialog management. By using client-side applications for such small tasks, we can reduce bandwidth utilization between the client and the server.

As described in Section 2, client-side applications can use the "back door" of the VoiceXML Interpreter to communicate with it asynchronously. This mechanism can be utilized if we want the system to interrupt a local (client-side) dialog as soon as it re-establishes a network connection with the remote server. However, it is also possible to implement substantial client-side applications simply by storing and accessing static VoiceXML documents within the client.

6. INITIAL PROTOTYPE SYSTEM

We have developed an initial prototype system that is integrated with the ScenarioXML and DialogXML compilers. Control of the dialog flow using grammars and lexicons as described in Section 4 has also been implemented; automatic grammar compilation from a corpus is currently being evaluated in a separate prototype system. The task switching described in Section 5 is also being tested in a separate prototype.

The initial prototype system was modified to communicate with an internal GPS module, as described in Section 2. Figure 5-7 shows a schematic diagram for the initial prototype system; the data flow sequence for a typical route guidance dialog is illustrated in detail. The GPS simulator, which provides functionality equivalent to a GPS module in a vehicle, plays the role of internal application as described in Fig. 5-1. The Route Planner is an example of an external database or information service, but it was implemented on the client side in the prototype because it requires frequent communication with the GPS simulator.

In this system, the user first asks the VoiceXML Interpreter to send an HTTP request, which includes the current position given by the GPS simulator and the destination given by the user's voice command. When the Dialog Manager receives the request, it asks the Route Planner how to get to the desired destination. The Route Planner provides directions in the form of a set of sentences, each of which is to be delivered to the user at a predetermined landmark point (e.g. intersection). Landmark point information is stored in the client system in order to align the dialog flow with the movement of the vehicle. The Dialog Manager combines the direction sentence set with the dialog scenario, which is pre-stored in ScenarioXML form; an appropriate VoiceXML document is created via intermediate compilation to DialogXML. As a result, the server can provide the client with a VoiceXML document that includes several turns of the dialog. The VoiceXML Interpreter receives the document, initiates the dialog, and then pauses after the first instruction is given to the user. When the vehicle approaches the next landmark point, the GPS simulator sends a trigger to the VoiceXML Interpreter via the asynchronous communication channel. Each time the VoiceXML Interpreter receives such a trigger, the next instruction is given to the user, and the Interpreter pauses and waits for the next trigger.

The interaction with the user may trigger other dialogs; for example, the user may inquire about parking facilities close to their desired destination. When the user triggers a new topic, the current dialog is suspended, and a new dialog is generated by the Dialog Manager. This new dialog will be aborted if a new trigger comes from the GPS simulator, and the suspended directions dialog will be re-activated so that the next instruction can be given to the user. To realize this capability, it is necessary for the system to recognize these asynchronous triggering events in all dialogs; fortunately, this does not require additional dialog development work on the part of the developer, because such triggering events can be defined as common arcs in the ScenarioXML representation.

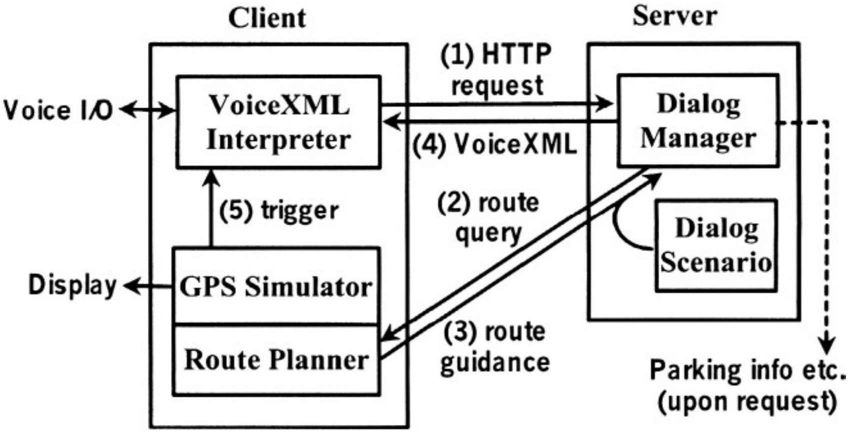


Figure 5-7. Schematic diagram of the initial prototype system.

From Fig. 5-7, we see that steps (1) to (5) represent the data flow sequence of a typical route guidance dialog. This architecture has a distinct advantage when compared to standard client-server dialog systems, since it can continue the route guidance dialog even if the communication between the client and the server is disconnected. In addition, the DM can provide dynamic information to the user whenever the connection to the remote server is established.

The new dialog (i. e. parking dialog) must be aborted by a trigger, because the only mechanism for embedding dialogs in VoiceXML is via the <subdialog> directive, which pushes the original dialog (e. g. directions dialog) down on a dialog stack and invokes a new dialog. As a result, when the user re-activates a previous (parking) dialog after the trigger was processed in the original dialog, the default VoiceXML action would be to push a new dialog and start it in its initial state (rather than returning to whatever the last active state was in a previously-activated dialog). However, if the server uses session variables to maintain a history of suspended dialogs and their last active state, it is possible to restart a dialog from the point at which it had been suspended. This capability represents an enhancement to our prototype system, and is under active development.

7. CONCLUSIONS

In this chapter, we described a dialog management architecture for car telematics systems. The architecture consists of a client and a server, and is designed to minimize the bandwidth of communications between them. On the server side, the Dialog Manager controls the current interaction with the user according to pre-defined scenarios written in ScenarioXML. The developer can define state-and-transition dialog scenarios using various predefined templates, which support integration of dynamic information from external databases and information services. Analysis of each user utterance within a dialog is achieved through application of a pre-selected grammar and lexicon, so that the developer has only to select appropriate sets of grammars and lexicons for each dialog state. These grammars may be written by hand, but it is also possible to construct them automatically using sample dialogs for each task. Finally, we described how the system switches control between the server and the client according to the current status of the communication channel. The system is robust in the presence of sudden network disconnections, and bandwidth utilization can be reduced by the use of client-side applications for simple tasks which do not require real-time access to dynamically changing information.

Our initial prototype system implements much of the architecture described above. Dialog scenarios are written in ScenarioXML, which combines the pre-defined dialog structure with dynamic information provided by the Route Planner in real time. The VoiceXML file includes several turns of the dialog, and the asynchronous communication channel of the VoiceXML Interpreter is used to advance the dialog in accordance with the vehicle's movement.

In the current system, context switching (between multiple active dialog instances) is realized by a push-and-pop style manipulation of a dialog stack. We are currently testing an extension of our architecture which uses session variables to enable switching between multiple (parallel) active dialogs.

ACKNOWLEDGEMENTS

The authors would like to thank Ichiro Akahori and Masahiko Tateishi of DENSO Research Laboratories for their support in collecting dialog corpora and assisting in the design and development of the initial prototype system.

REFERENCES

[1] W3C, "Voice Extensible Markup Language (VoiceXML) Version 2.0 Working Draft," http://www.w3c.org/TR/voicexml20/

[2] B. Carpenter, S. Caskey, K. Dayanidhi, C. Drouin, and R. Pieraccini, "A Portable, Server-Side Dialog Framework for VoiceXML," Proc. of International Conference on Spoken Language Processing, 2002

[3] R. Pieraccini, S. Caskey, K. Dayanidhi, B. Carpenter, and M. Phillips, "ETUDE: A Recursive Dialog Manager with Embedded User Interface Patterns," Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding, 2001

[4] E. Nyberg, T. Mitamura, P. Placeway, and M. Duggan, "DialogXML: Extending VoiceXML for dynamic dialog management," Proc. of Human Language Technology Conference, 2002

[5] Sun Microsystems, "JavaServer Pages," http://java.sun.com/products/jsp/

[6] X. Huang, F. Alleva, H. Hon, M. Hwang, K. Lee, and R. Rosenfeld, "The SPHINX-II Speech Recognition System: An Overview," Computer Speech and Language," vol.2, pp.137-148, 1993

[7] E. Nyberg, and T. Mitamura, "The KANTOO Machine Translation Environment," Proc. of AMTA-2000

[8] T. Kujirai, H. Takahashi, A. Amano, and N. Hataoka, "Development of VoiceXML Interpreter and Continuous Words Recognition Engine - Development of Speech Recognition Technologies for Voice Portal," (in Japanese) IPSJ SIGNotes, SLP-33-12, 2000

[9] M. Tateishi, I. Akahori, S. Judy, Y. Obuchi, T. Mitamura, and E. Nyberg, "A Spoken Dialog Corpus for Car Telematics Services," Proc. of Workshop on DSP in Vehicular and Mobile Systems, 2003

[10] S. M. Shieber, H. Uszkoreit, J. Robinson, and M. Tyson, “The formalism and Implementation of PATR-II,” SRI International, Menlo Park, California, 1983.

[11] A. Black, “Finite State Machines from Feature Grammars,” Proc. of Int. Workshop on Parsing Technologies, 1989

Chapter 6

USE OF MULTIPLE SPEECH RECOGNITION UNITS IN AN IN-CAR ASSISTANCE SYSTEM[1]

Alessio Brutti[1], Paolo Coletti[1], Luca Cristoforetti[1], Petra Geutner[2], Alessandro Giacomini[1], Mirko Maistrello[1], Marco Matassoni[1], Maurizio Omologo[1], Frank Steffens[2], Piergiorgio Svaizer[1]

[1]*ITC-irst (Centro per la Ricerca Scientifica e Tecnologica), I-38050 Povo - Trent, Italy;* [2]*Robert Bosch GmbH, Corporate Research and Development, P.O. Box 10 60 50, Stuttgart Germany.* *Email: brutti@itc.it*

Abstract: This chapter presents an advanced dialogue system based on in-car hands-free voice interaction, conceived for obtaining driving assistance and for accessing tourist information while driving. Part of the related activities aimed at developing this "Virtual Intelligent Codriver" are being conducted under the European VICO project. The architecture of the dialogue system is here presented, with a description of its main modules: Front-end Speech Processing, Recognition Engine, Natural Language Understanding, Dialogue Manager and Car Wide Web. The use of a set of HMM recognizers, running in parallel, is being investigated within this project in order to ensure low complexity, modularity, fast response, and to allow a real-time reconfiguration of the language models and grammars according to the dialogue context. A corpus of spontaneous speech interactions was collected at ITC-irst using the Wizard-of-Oz method in a real driving situation. Multiple recognition units specialized on geographical subdomains and simpler language models were experimented using the resulting corpus. This investigation shows that, in presence of large lists of names (e.g. cities, streets, hotels), the choice of the output with maximum likelihood among the active units, although a simple approach, provides better results than the use of a single comprehensive language model.

Keywords: Automatic speech recognition, in-car dialogue system, driving assistance, language models.

[1] This work was partially funded by the Commission of the EC, Information Society Technologies (IST), 2000-25426, under VICO. Partners of the VICO project are Robert Bosch GmbH (D), DaimlerChrysler AG (D), ITC-irst (I) and Phonetic Topographies NV (B).

1. INTRODUCTION

The application of telematics in the car environment involves the integration of onboard computer, onboard devices, global positioning and wireless communication systems.

As a safe, reliable and comfortable interaction with these systems is of particular relevance while driving, Automatic Speech Recognition (ASR) technology in the car environment has gained more and more interest for the emerging automotive applications appearing on the market.

Robustness and flexibility of hands-free ASR systems in adverse environment are still challenging topics of research [1]-[4]. Speech signals acquired by hands-free systems on a moving car are generally characterized by low SNR and are affected by various sources of corruption. Engine and tyres contribute mainly low frequency noise, while aerodynamic turbulence, predominant at high speed, has a broader spectral content. Other noise components are unstationary and unpredictable (e.g., road bumps, rain, traffic noise, etc.).

A further reduction of the speech recognizer accuracy is caused by acoustic effects of the car enclosure, spontaneous speech phenomena and speaking style modifications (i.e. Lombard effect), especially in conjunction with the word confusability induced by large vocabularies.

The European project VICO (Virtual Intelligent CO-driver) has the goal of developing an advanced in-car dialogue system for the vocal interaction in natural language with an agent able to provide services as navigation, route planning, hotel and restaurants reservation, tourist information, car manual consultation [5],[6]. The planned system includes a robust hands-free speech recognizer, connected with a natural language understanding module allowing for spontaneous speech interaction and an advanced and flexible dialogue manager able to adapt itself to a wide range of dialogue situations. A further module constitutes the interface for a dynamic information retrieval and an efficient data extraction from databases containing geographic and tourist information. Voice interaction can be in English, German or Italian. ITC-irst has in charge the development of the ASR engine for Italian, while the corresponding engines for English and German are developed by Daimler Chrysler AG [7].

All the modules are integrated into a CORBA system architecture, and a common interface was specified to connect the recognizers of different languages to the same natural language understanding module. Due to the need of alignment among language models for the different languages as well as to the need of reducing the complexity while managing large vocabularies (e.g., lists of streets and points of interest in a city, cities in a region, etc.), a framework was realized, based on the concept of several speech recognition units that run in parallel and use class-based statistical language models or grammars.

The objective of this chapter is that of investigating on a simple selection method to choose the most likely output among those provided by a set of recognition units fed with a common input signal [8]. A corpus of real spontaneous speech utterances acquired in the car is employed to test the accuracy of the resulting speech recognizer. The chapter is organized as follows: section 2 introduces the general system architecture and presents some details about the principal subsystems; section 3 describes the test database collected through Wizard-of-Oz (WOZ) and some experiments with multiple recognition units. In the final section, we draw some conclusions and describe future developments.

2. SYSTEM ARCHITECTURE

The general architecture of the VICO system is shown in Figure 6-1, where the blocks “Front-end processing”, “Recognition engine” and “Recognizer output selector” constitute the subsystem used in the experiments described later in this chapter.

The front-end processing is based on robust speech activity detection, noise reduction and feature extraction. The recognition module is conceived as a set of Speech Recognition Units (SRU) working in parallel, each one with its own specialized Language Model (LM), followed by an output selection module. The aim of this configuration is that of looking for a more reliable input to the Natural Language Understanding (NLU) module, than what would be obtained when using a single comprehensive Language Model (LM) and a related very large vocabulary.

As shown in the figure, we assume that the Dialogue Manager (DM) can dynamically load new LMs and activate or deactivate the single recognition units at each dialogue step (i.e. recognition process) according to the context of the dialogue interaction. If no one of the outputs of the units is judged reliable, the DM can load new LMs and ask for a further recognition step on the given input utterance.

Note that the SRUs, once loaded, can be selected to be running at the same time, which means that a user utterance is being processed in parallel by all active SRUs in a very efficient manner, this way avoiding the delay that would be introduced by any equivalent sequential recognition approach.

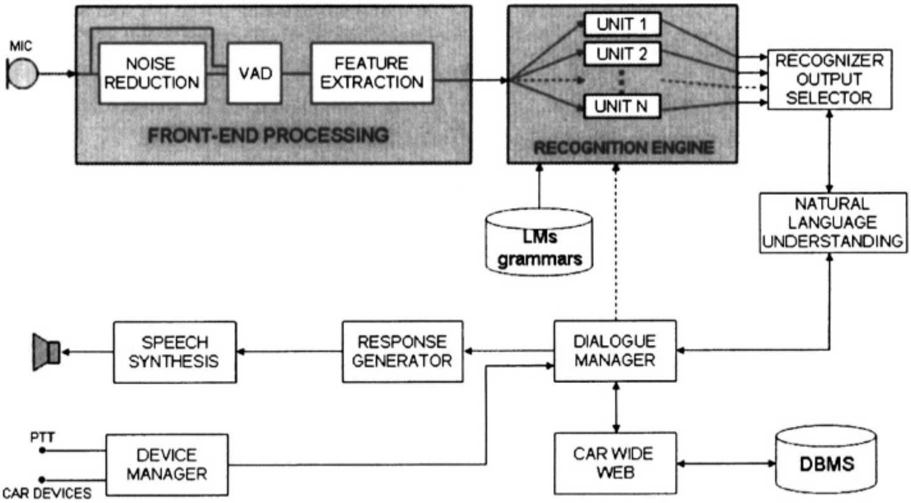


Figure 6-1. System Architecture.

The diagram also shows the other modules of the VICO system. It is worth noting that the Car Wide Web (CWW) module is an interface to a set of databases using an XML-Schema based protocol to communicate with the DM module. Presently, it allows a fast access to a tourist database that includes most of the relevant information about the italian Trentino region (produced by Azienda per la Promozione Turistica del Trentino) and to a geographical-topographical database (produced by TeleAtlas) for any query concerning navigation.

The dialogue between the system and the user is based on information resident in static or dynamic databases. A static database may contain historical information regarding a city being visited, a dynamic database (through an Internet connection) may contain weather news or could allow a hotel reservation. CWW primary task is to understand the queries coming from the DM and to retrieve the data from the databases. The actual connection to the data is delegated to a specific API, so CWW maintains a certain independence from the physical structure of the databases.

2.1 Front-end processing

Basic noise reduction algorithms are an easy and effective way to reduce mismatch between noisy conditions and clean HMMs, and can also be used with some benefits in matched conditions, as was shown in [9]. On the basis of that work, magnitude spectral subtraction and log-MMSE estimation were adopted for background noise reduction, together with quantile noise estimation.

In [10] an optimal set of parameters was determined for the use of spectral subtraction and log-MMSE on a connected digit recognition task; the same set is used here. The noise subtraction module is used only for far-microphone input processing. The front-end processing includes a Voice Activity Detection (VAD) module. It is based on the energy information in the case of close-talk input and on a spectral variation function technique applied to the output of the Mel-based filter bank in the case of far-microphone signal. According to preliminary experiments on SpeechDat.Car material [11], both techniques allow recognition performance equivalent to that determined by using manually segmented utterances, except for cases of unstationary noise events.

The feature extraction module processes the input signal pre-emphasizing and blocking it into frames of 20 ms duration from which 12 Mel scaled Cepstral Coefficients (MCCs) and the log-energy are extracted. MCCs are normalized by subtracting the current MCC means. The log-energy is also normalized with respect to the current maximum energy value. The resulting MCCs and the normalized log-energy, together with their first and second order time derivatives, are arranged into a single observation vector of 39 components.

2.2 Recognition Engine

The recognition engine for the Italian language is composed of a set of standard HMM recognition units. Each of them runs independently and processes the features provided by the front-end module. The HMM units are based on a set of 34 phone-like speech units. Each acoustic-phonetic unit is modeled with left-to-right Continuous Density HMMs with output probability distributions represented by means of mixtures having 16 Gaussian components with diagonal covariance matrices. HMM training is accomplished through the standard Baum-Welch training procedure. Phone units were trained by using far-microphone (and close-talk) signals available in the Italian portion of SpeechDat.Car corpus. The training portion of this corpus consists in about 3000 phonetically rich sentences pronounced by 150 speakers.

A crucial aspect is the selection of the output to feed NLU module. For a given input utterance, the outputs provided by the different active units have to be compared each other in a reliable way. Although more sophisticated approaches are possible [12], in this work the simplest decision policy is adopted, which is based on the selection of the output having the maximum likelihood.

2.3 Natural Language Understanding

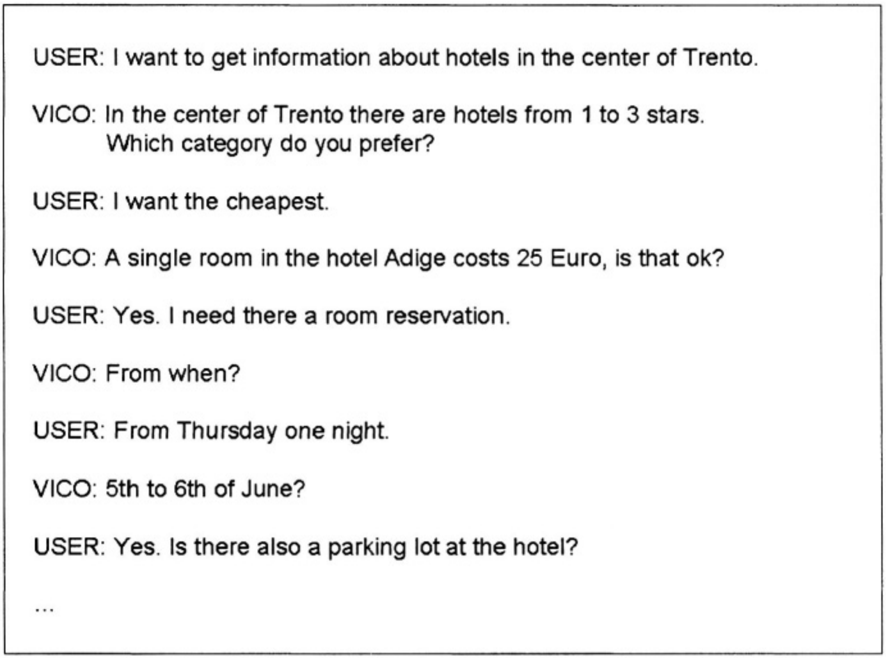
The Natural Language Understanding (NLU) module (being developed at Daimler-Chrysler AG) has to take care of the spontaneous speech input and possible incomplete or ungrammatical sentences. The parser must be able to deal with all these factors and provide a semantic representation of the user's sentence. The whole structure of an utterance is not currently checked for consistency, rather single elements containing the phrase have to be identified. The resulting semantic representation is composed by a set of slots instantiated by the meaningful words of the utterance, taking into account the context of the on-going dialogue.

2.4 Dialogue Manager

At the core of the spoken dialogue system, the Dialog Manager aims at reacting appropriately to the user's spontaneous requests in all circumstances. The input to this module is the semantic information delivered by the NLU; the output consists of queries to the databases through the CWW module, instructions to the Response Generator, and the selection of grammars for the language understanding module according to the dialogue status. Basically, the DM must be able to identify the current task requested by the user and to obtain the information needed to fulfill it. This has to be done also in case of incomplete, ambiguous or erroneous input, to integrate or clarify the information provided by the driver, and to accept a request of modification in a possible negotiation step.

The implemented DM is based on a description of the application, which is basically characterized by a set of contexts, each one containing some semantically relevant concepts, and by a set of procedures that describe the actions to be performed during the dialogue. The goal of the DM is to fill the concepts of the active context in a consistent way, then to require some external actions and finally to move to another context.

The Dialog Manager should be aware of the profile of the user, retrieved via the Car Wide Web. The user profile should contain information about user's preferences and habits, like preferred hotel chains, restaurants, familiarity with the system. For example DM should be able to customize the interaction in order to avoid to be petulant with expert users or to be too concise with novice users. A sample dialogue is reported in Figure 6-2 as an exemplification of the typical interaction between the driver and the virtual co-driver.



USER: I want to get information about hotels in the center of Trento.

VICO: In the center of Trento there are hotels from 1 to 3 stars.
Which category do you prefer?

USER: I want the cheapest.

VICO: A single room in the hotel Adige costs 25 Euro, is that ok?

USER: Yes. I need there a room reservation.

VICO: From when?

USER: From Thursday one night.

VICO: 5th to 6th of June?

USER: Yes. Is there also a parking lot at the hotel?

...

Figure 6-2. Sample dialogue.

2.5 Car Wide Web

The Car Wide Web is the module responsible for the retrieval of the data needed by the system. Almost all the dialogue that takes place between the system and the user is based on information resident in databases, which can be static or dynamic. A static database may contain historical information regarding a city being visited; a dynamic database (through an Internet connection) may contain weather forecasts or could allow a restaurant reservation. CWW primary task is to understand the queries coming from the DM and to retrieve the data from the databases. The actual connection to the data is delegated to a specific API (see Section 2.6), so CWW maintains a certain independence from the physical structure of the databases.

In order to have a flexible and standard formalism to exchange the data between CWW and DM, all the queries and responses are wrapped in an XML format. It has been decided to define two XSDs, one for the data going to the CWW and another for the data returning back to the DM. XSDs can be seen as grammars defining the format of XML files by putting some restrictions on the structure of the files and on the values contained. Using such a formalism, it comes easy to discover errors and missing data in both the structure and the content of the requests. This leads to an efficient error handling that avoids wrong calls to the database's API [13].

Typical data retrieval tasks fulfilled by the currently implemented CWW are:

- *Hotel Information Retrieval:* Given some constraints, all the suitable hotels of a certain area are returned, each one with all the information available (prices for single/double room, phone number, stars, full address, available services, parking and restaurant inside).

- *Hotel and Restaurant Reservation:* Once a hotel has been identified, rooms can be reserved for a certain number of days. A reservation can be done even for the restaurant, if the hotel has one inside. Reservations are simulated at the moment since a real Internet connection with the Hotels is not yet available.

- *Point Of Interest (POI) Retrieval:* Given a POI type (i.e. petrol station, museum, etc.) and specified a certain area, all the matching POIs present are returned, each one with all the information available (currently the full address).

- *Simple Route Query:* This query is the fundamental query for the route planning. A given address (even incomplete) is checked for its existence. Possible inconsistencies and ambiguities are reported to the DM in order to take a suitable action and interrogate the user.

2.6 Databases and API

The data used by the Dialog Manager may come from static or dynamic databases. Static data needed during the interaction are stored in a MySQL database installed in the in-car PC. This database includes information on hotels (name, hotel type, hotel chain, stars, number of rooms, prices, phone number, services offered, complete address, hotel surroundings) and on POIs (name, brand name for special cases like petrol stations, complete address). The geographic part of the database includes streets, places, municipalities, provinces and countries, organized in a hierarchical structure in such a way that every lower order entry is contained in one higher order entry (each place is contained in one municipality, which is contained in one province). These geographic entries are linked with the tourist part of the database using special codes to easily identify the position of each hotel and each POI in the geographic hierarchy and to quickly respond to conditional queries (such as "*I'm looking for a hotel X in the municipality Y*"). Both tourist and geographic data have been geocoded, their entries contain information on position coordinates to allow the introduction of a navigation system in the prototype.

The API developed to access the database is able to check the addresses for consistency and uniqueness. When it receives an address as a 5-element object (province, municipality, place, street, street number), it automatically checks whether there are zero, one or more than one correspondence in the database. In the first case, which means that the system is receiving an inconsistent request from the user, it incrementally drops elements and tries to redo the query with fewer restrictions until it finds at least one item. In this way the API is able to return to CWW a suggestion on what may be the wrong element in the query. This will be returned to the Dialog Manager that will take an opportune action with the user.

Foreseen improvements of the database content include:

- insertion of phonetic transcription for names and insertion of multiple names. The database will contain phonetic transcriptions in three languages in order to let the system dynamically build speech recognition grammars. Moreover, it will have multiple names of hotels, POIs and streets, to deal with the problem of users that say only a part of the name or that use different names for the same POI.
- insertion of descriptive information on POI. This information will be structured in different levels of detail. Starting from a general description of the POI, the user will be given the possibility to obtain more specific information. For example, in the case of a castle, the first description could be a general single sentence on the castle that offers the user the possibility to ask for the history, the art or the architecture of the building. These topics correspond in the database to more specific descriptions that in turn will lead to even more detailed ones, such as the construction of the castle, its middle age history, its modern day history. In this way the user

is not overwhelmed with synthesized information of a POI but may explore only the desired information.

- POIs will be enriched with information about opening hours, tickets and phone numbers.

Other functionalities are planned to expand the current prototype as follows:

- the database will contain the car manual, hierarchically organized to allow consultation via a speech interaction.
- when the Internet connection will be available, the database will contain a dynamic part which automatically downloads news from various web sites and which is able to show the user the latest news according to his/her preferences (finance, weather, politics, etc.).
- all the information that concerns user habits and preferences will be stored in a user profile database and loaded by the Dialog Manager during dialogue start-up.

3. EXPERIMENTAL SETUP

Some recognition experiments were conceived to evaluate the convenience of using multiple recognition units in order to increase the system performance without reducing the language coverage. Another aspect under investigation is the reliability of the sentence likelihood as score for the selection of the most promising recognized sentence among the unit outputs.

Data were collected with the "Wizard of Oz" (WOZ) technique. This method allowed to acquire speech of real drivers using an apparently fully functioning system (the *codriver*), whose missing recognition capabilities were supplied by an hidden human operator (the *wizard*).

3.1 WOZ Database

An Italian WOZ-based data collection was organized in order to reconstruct a real situation in which the driver tries to fulfill, by voice interaction, tasks as:

- Reach a Point-Of-Interest (POI) in Trento city
- ask for hotel/restaurant information and book a room or reserve a table
- ask information about the car
- ask information about a museum, a church, etc.

These represent the typical scenarios taken into consideration by the VICO project.

During recordings, a co-driver was always in the car to describe each goal the driver had to pursue by voice interacting with the system. The wizard was at ITC-irst labs, connected to the mobile phone of the car. A specific setup was designed in order to simulate an interaction as realistic as possible and to allow a synchronous speech acquisition through two input channels, one connected to a close-talk head-mounted microphone (denoted as "CT") and the other to a far-microphone placed on the ceiling (denoted as "Far"). The audio prompts were produced by using a commercial text to speech synthesizer.

The present release includes 16 speakers (8 males + 8 females), that uttered a total of 1612 spontaneous speech utterances (equivalent to 9150 word occurrences). The total speech corpus duration is 132 minutes (mean duration of utterance is 4.9 sec) and the total vocabulary size is 918 words.

Note that all of the speakers were naive to the use of this type of systems and that the wizard behavior was based on an interaction model, previously defined, that comprised the simulation of recognition errors typical of the foreseen real scenario. As a result, many sentences include typical spontaneous speech problems (e.g. hesitations, repetitions, false starts, wrong pronunciations, etc.) and often consist in many words (in a few cases the input utterance contained more than 25 words). The realism of the experiment is also shown by the fact that at the end of the experiment, after more than one hour, all the speakers declared they were not aware of the fact that a human was interacting with them.

3.2 Recognition experiments

The present architecture is based on parallel recognizers covering distinct application domains and/or geographical clusters. The baseline performance, shown in Table 6-1, is evaluated using a single class-based language model, trained on a corpus of about 3000 sentences that cover different applications domains such as navigation, hotel reservation, address book management, questions about the car. The geographic coverage of this LM, indicated by the suffix *Cgl*, is the whole Trentino province, including names of cities, streets, hotels, restaurants, POIs (churches, castles, museums). Equal probability has been assigned to all the items within each geographical class. The derived LM includes about 12000 words and has a Out-Of-Vocabulary (OOV) rate (evaluated on the WOZ data) of 1.1 %.

LM	Vocabulary size	OOV Rate	WRR (CT)	WRR (Far)
Dgl-Cgl	12k	1.1%	58.8%	46.1%

Table 6-1. Baseline performance for the CT and Far input channels.

Here and in the following the various language models are named according to the domain (denoted by letter D) covered by data used in the training phase. If a LM contains geographical classes, its name includes information about the cluster (denoted by letter C) which contributes the lists of names (cities, streets, hotels, etc.) used to expand the classes. Therefore, for example, *Dgl-Cgl* denotes the LM trained on the global (*gl*) domain with classes expanded with the global (*gl*) lists of names.

There are different options for building smaller LMs that contribute to provide the complete coverage of the application domains foreseen in the VICO system. A simple solution is to reduce the contents of the classes associated to the large lists (cities, streets, hotels, etc.) introducing some geographic clusters and building several LMs, each one covering only a reduced area: in our setup Trentino has been divided in 7 geographic clusters *(C1,C2,C3,C4,C5,C7*).

Another possible strategy in order to exploit different recognition units is to build LMs not containing the classes associated to the big lists. This idea derives from the observation than a generic dialogue contains a relatively low number of sentences including the pronunciation of a noun associated to a big list: this leads to the introduction of 2 further small LMs, namely *Dge* and *Dcmd,* that have been trained removing from the corpus the sentences with geographic class contents (e.g. cities, streets, hotels, POIs). In particular *Dcmd* is a very restricted LMs (the vocabulary size is 130) and it should handle only confirmation/refusal expressions and short commands to the system.

Table 6-2 shows the results for these new LMs: *Dgl* denotes the original global LM while the suffix *Ci* specifies the geographic cluster covered. The higher WRRs obtained with *Dgl-C1* are motivated by the fact that the WOZ material regards geographic items mainly associated to *C1*, i.e. the Trento city area, where the acquisition took place. It is worth mentioning that although WRR of *Dge* and *Dcmd* is rather low, the relative string recognition rate shows that these LMs cover adequately a relevant part of the corpus.

LM	Vocabulary Size	OOV rate	WRR (CT)	WRR (Far)
Dgl-C1	3.0 k	2.7%	62.5%	55.9%
Dgl-C1	2.5 k	4.6%	58.1%	50.6%
Dgl-C1	3.5 k	4.6%	55.2%	46.2%
Dgl-C1	3.1 k	4.8%	57.4%	50.3%
Dgl-C1	3.8 k	4.5%	57.8%	49.5%
Dgl-C1	4.2 k	4.8%	57.4%	49.9%
Dgl-C1	4.8 k	4.2%	57.4%	49.0%
Dge	1.3 k	16.7%	46.2%	40.2%
Dcmd	130	65.6%	12.4%	8.3%

Table 6-2. Vocabulary size, OOV and recognition results on the overall WOZ corpus for the restricted LMs.

A critical issue related to the multiple units approach is how to select the most reliable output on the basis of a confidence score. Anyway, even adopting the simplest strategy of selecting the output string on the basis of its likelihood (ML), a considerable improvement has been observed, as shown in Table 6-3. Moreover, once the recognition engine is integrated on the complete system, the DM is in principle able to predict the most likely domains of the following interaction step, or to assign a proper weight to the members of the geographical classes. Only a few recognition units should therefore remain active, on the basis of the DM's prediction.

	WRR (CT)	**WRR (far)**
Dgl-Cgl	58.8%	46.1%
ML Dgl-C[1-7]	62.7%	51.4%
ML Dgl-Cgl+Dcmd+Dge	61.4%	49.2%
ML Dgl-C[1-7]+Dcmd+Dge	64.2%	53.9%

Table 6-3. Recognition results in the multiple units framework: the maximum likelihood criterion (ML) selects the most promising output among the different recognized strings.

4. CONCLUSIONS AND FUTURE WORK

This work represents a preliminary step in the development of a dialogue system for in-car voice interaction with advanced services of navigation assistance and tourist information access. As the system complexity in the given framework is a crucial aspect, our research focuses on the development of multiple fast recognition units and a suitable combination strategy that may lead to better performance than the adoption of a single full-coverage recognizer.

Even if the application of a maximum likelihood criterion to select the recognition output represents the simplest choice, it offers some advantages in terms of performance and also complexity, assuming the ability of the dialogue manager to predict at each interaction step the most likely domain in terms of geographic area as well as of dialogue context. The geographic clustering seems to be effective in presence of large list of names (cities, streets and hotels names) that give rise to a considerable acoustic confusability. Work is under way for what regards the selection of more reliable outputs, on the basis of confidence measures and word hypotheses graphs.

REFERENCES

[1] Proceedings of the Hands-Free Speech Communication Workshop (HSC), Kyoto (Japan), 2001.

[2] M. Omologo, P. Svaizer, M. Matassoni, "Environmental conditions and acoustic transduction in hands-free speech recognition", Speech Communication, vol.25, pp. 75-95, 1998.

[3] J.H.L. Hansen, X. Zhang, M. Akbacak, U. Yapanel, B. Pellom, W. Ward, "CU-Move: Advances in In-Vehicle Speech Systems for Route Navigation", Proc. of the Workshop on DSP in Mobile and Vehicular Systems, Nagoya, Japan, 2003.

[4] N. Kawaguchi, S. Matsubara, I. Kishida, Y. Irie, Y. Yamaguchi, K. Takeda, F. Itakura, "Construction and Analysis of the Multi-layered In-car Spoken Dialogue Corpus", Proc. of the Workshop on DSP in Mobile and Vehicular Systems, Nagoya, Japan, 2003.

[5] P. Geutner, F. Steffens, D. Manstetten, Design of the VICO spoken dialogue system: evaluation of user expectations by Wizard-of-Oz experiments , Proc. of LREC, Las Palmas (Spain), 2002.

[6] P. Coletti, L. Cristoforetti, M. Matassoni, M. Omologo, P. Svaizer, P.Geutner, F. Steffens, "A speech driven in-car assistance system", Proc. of the IEEE Intelligent Vehicles [IV 2003], Columbus, OH, 2003.

[7] H. Hüning, A. Berton, U. Haiber, F. Class, "Speech Recognition Methods and their Potential for Dialogue Systems in Mobile Environments", ISCA Workshop, Kloster Irsee (Germany), June 2002.

[8] L. Cristoforetti, M. Matassoni, M. Omologo, P. Svaizer, "Use of parallel recognizers for robust in-car speech interaction", IEEE ICASSP-03: Inter. Conf. Acoustics, Speech and Signal Processing, Hong Kong, 2003.

[9] M. Matassoni, G.A. Mian, M. Omologo, A. Santarelli, P. Svaizer, "Some experiments on the use of one-channel noise reduction techniques with the Italian SpeechDat Car database", Proc. of ASRU, Madonna di Campiglio (Italy), 2001.

[10] M. Matassoni, M. Omologo, A. Santarelli, P. Svaizer, "On the joint use of noise reduction and MLLR adaptation for in-car hands-free speech recognition", IEEE ICASSP-02: Inter. Conf. Acoustics, Speech and Signal Processing, Orlando (FL), 2002.

[11] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, J. Allen, "A Large Speech Database for Automotive Environments", Proc. of LREC, Athens (Greece), 2000.

[12] R. Solsona, E. Fosler-Lussier, H.J. Kuo, A. Potamianos, I. Zitouni, "Adaptive Language Models for Spoken Dialogue Systems", Proc. of ICASSP, Orlando (FL), 2002.

[13] P. Coletti, L. Cristoforetti, M. Matassoni, M. Omologo, P. Svaizer, "Developing a speech interaction system for the car", Proc. of the 8th International Conference on Human Aspects of Advanced Manufacturing: Agility & Hybrid Automation [HAAMAHA 03], Rome, Italy, 2003.

This page intentionally left blank

Chapter 7

HI-SPEED ERROR CORRECTING CODE LSI FOR MOBILE PHONE

Yuuichi Hamasuna[1], *Masayasu Hata*[2], *Ichi Takumi*[3]
[1]*DDS Inc., Otohbashi, Nakagawa-ku, Nagoya 454-0012 Japan;* [2]*Chubu University, Kasugai, Aichi 487-8501 Japan;* [3]*Nagoya Institute of Technology, Gokiso, Nagoya, Aichi 466-5588 Japan.* *Email:hamasuna@dds.co.jp*

Abstract: In recent years, the transmission speed of a cellular phone of the next generation has reached 100Mbps, and the transmission speed of optical communication amounts to 40 Gbps. Accordingly, demand for robust error correction code with high-speed processing of a Gbps class is increasing. The proposed code "High dimensional torus knot code" performs well in a field with many errors. In a performance comparison with the Reed-Solomon code, the performance of the proposed code is better than the Reed-Solomon code in an environment with 10^{-1}-10^{-2} error. Moreover, doing a simulation in a CDMA communication environment, fluttering of error property has not occurred with the product code of convolutional codes (as inner code: the rate is 1/2) and the proposed code (as outer code: the rate is 0.53). Alternatively, under the same conditions, a fluttering error occurred in the Turbo cord. By applying the LSI technology, we developed ASIC of the proposed code, and FPGA for the high-speed MPEG communication device. We developed the three-dimensional, size-nine 3Dm9 and 4Dm5 chip. More specifically, the 3Dm9-code chip (developed in 2001) having a rate of 0.70 and block length of 729 bits was burnt onto a 100-kilogate, 0.35-micron-order LSI chip, and the 4Dm5-code chip (r=0.41, block=625, developed in 1999) was burnt onto a 50-kilogate, 0.6-micron-order LSI chip. Moreover, the 3Dm9-code chip was operated at a clock speed of 66.6MHz with throughput of 48Gbps. Finally, after applying the developed FPGA, the high-speed MPEG communication device can transmit a movie signal of 33Mbps.

Keywords: High-Dimensional Discrete Torus Knot Code, Robust error correction code, High-throughput(48 Gbps) Coder and Decoder, Wired logic

1. INTRODUCTION

Recently, transmission speed of radio communications represented by LAN and cellular phones has largely increased. Hence, there is an increasing demand for robust and high-speed error correction code. As for the high-speed and robust error correction code, the Reed-Solomon code is mentioned first, but its correction capacity is inferior to the proposed code in an environment with an error of 10^{-1} to 10^{-2}. Moreover, the Turbo code does not perform well in a practical environment.

In order to meet requirements, we have proposed a topological new code "High dimensional torus knot code." The proposed code is resistant to a random error of 10^{-1} to 10^{-2}, and to the burst errors. This code can be realized as a high-speed circuit, which makes full use of parallel operation and wired logic technology, because it consists of simple parity operations. We successfully realized the hardware implementation of the proposed code on an ASIC and FPGA with throughput of 6 to 48 Gbps, and we developed the high-speed MPEG communication device by applying the proposed code. The proposed code will be expected to work well in degraded channel situations such as cellular phone [1].

2. ARCHITECTURE OF HIGH DIMENSIONAL DISCRETE TORUS KNOT CODE

Figure 7-1 represents a schematic diagram of the proposed code that shows data flow from input to output. The proposed code consists of two processing blocks, which are a high-dimension parity code and a torus knot scramble.

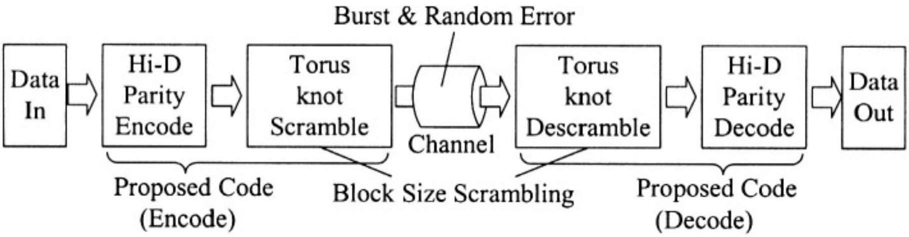


Figure 7-1. Schematic diagram of proposed code.

The high-dimension parity code has an n dimensional discrete cubical structure in Figure 7-2. Each dimensional axis has a size of m, and the axis consists of m code points. The code is denoted as nDm, where n is dimension and m is size. For example, 3Dm5 means that dimension of the code is three and size is five. Each axis has one single redundant digit and m-1 data digits, which satisfies the even parity. There are n such independent parity axes, which form m^n code points. The nDm code has $(m-1)^n$ data digits, totaling m^n digits including parity redundant digits, and has a transmission rate of $R=(1-1/m)^n$. On the decoding side, each digit is checked by n parity check lines and is corrected when the number of failed parity check lines exceeds the threshold value.

On these code points, the transmission order runs obliquely to form a discrete torus knot similar to the number change in Figure 7-3. Therefore, the errors that appear on the block are uniformly distributed on each code axis.

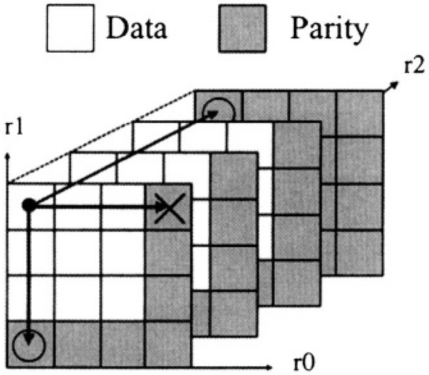


Figure 7-2. Hi-dimension parity code, an example of 3Dm4.

3. PERFORMANCE OF THE PROPOSED CODE

Figure 7-4 illustrates the results of the proposed code to the Reed-Solomon code. Simulation conditions are summarized in Table 7-1. Although the performance of the Reed-Solomon code is higher than the proposed code in an area with few errors, but in an area with many errors, the performance of the proposed code is higher than the Reed-Solomon code.

When the proposed code is constituted into the high dimension structure as four dimensions, the number of uncorrectable patterns extremely decreases. This is the reason why the performance of the proposed code is better than the Reed-Solomon code in an area with many errors. An error of 10^{-1} from 10^{-2} encounters frequently in the radio communications.

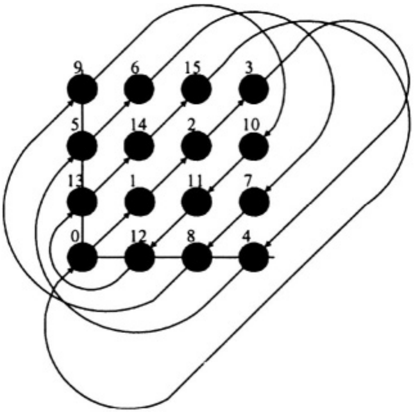


Figure 7-3. Transmission order (Torus scramble), an example of 2Dm4.

Input error type	Random error generated by 512 bits M-sequence

Code Type	Block size	Rate
Reed-Solomon (255,159,8)	2040 bit	0.62
Reed-Solomon (255,203,8)	2040 bit	0.79
Proposed code (4Dm9)	6551 bit	0.62
Proposed code (4Dm18)	104976 bit	0.79

Table 7-1. Simulation parameters.

A radio communications in a cellular phone is a degraded channel. Accordingly, the proposed code's performance in CDMA, which is the communication system of a cellular phone of the current generation, was evaluated using simulator. The tool used for the simulation was a MATLAB, and the CDMA environment was built using the attached IS-95 library in the MATLAB. The proposed code has a disadvantage in radio communications because it decodes by hard decision. In analog channel, introducing a soft-decision code can raise a correction ability. When we used the product code, an outer code was the proposed code, and an inner code was the convolutional code. Simulation conditions are summarized in Table 7-2. Figure 7-5 indicates that a fluttering error occurred in the characteristic curve of the Turbo code, and the Turbo code did not reduce the error to the 10^{-5}. Alternatively, the proposed code made a 10^{-7} reduction in error.

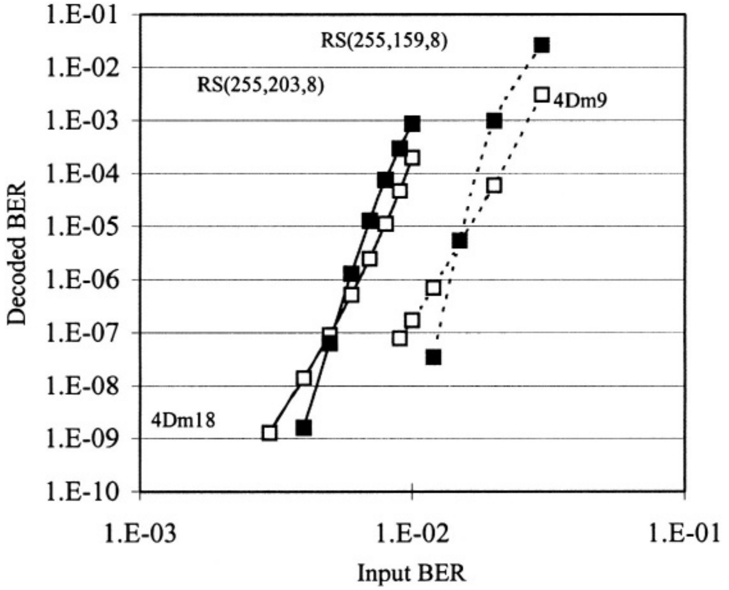


Figure 7-4. Results of comparing reed-solomon code with the proposed code.

Figure 7-6 shows the improvement in the decoded bit error rate versus Eb/No for AWGN environment. At the decoded BER of 10^{-5}, the required input BER of the proposed code is 0.020 and 0.026 for 4Dm6 and 4Dm5, respectively. In contrast, the coding gain of the 8-PSK TCM is given by the minimum Euclidean distance of $\left\{2^2+2+\left(\pi/4\right)^2\right\}^{1/2}$=2.141(6.61dB) against the distance of $\sqrt{2}$(3.01dB) of uncoded 4-PSK. Therefore, the required Eb/No of the proposed code with 8-PSK TCM is estimated as 2.0 to 2.5dB for 4Dm5 and 4Dm6 codes to obtain 10^{-5}. The value is sufficiently competitive to that of the Turbo code with interleaver of 1024, especially for high-grade BER ranges less than 10^{-7}. The performance is also similar to the concatenated code of the Reed-Solomon (204,188) with the 3/4 punctured convolution code for digital TV and for space communications [2,3,4,5].

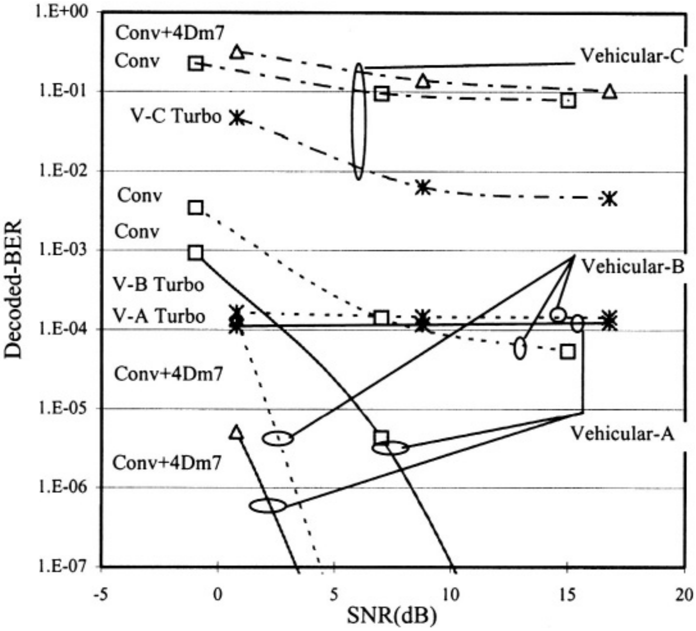


Figure 7-5. Results of the simulation in CDMA environment.

4. HARDWARE IMPLEMENTATION

The proposed code consists of a simple parity check code. Therefore, we discovered the simple relation between parity and data digits. Accordingly, a significant part of the encoding and decoding processes are substituted with wired connections between memory cells. We developed a program to automatically generate a VHDL source program, adjusting to its code dimension and size. VHDL is a hardware description language, which translates software programs into hardware configurations. Installation of hardware circuitry through software language is especially suitable for the circuitry that consists of repetition or regularity. By applying this program, it is possible to generate the encoder and decoder for three to five dimensional codes.

Communication Parameter	CDMA(IS-95)Standard Modulation Type:BPSK/QPSK Speed:9.6kbps Defuse code:LongPN,64Walsh, PairPN Defuse rate:128
Input error type	AWGN+Multipath-fading (Used Model is Vehicular-A,B,C, Doppler frequency 200Hz)

Code Type	Rate
Convolutional code (k=9,G (753,561) ,8bit soft decision)	1/2
Turbo Code (iteration6) Convolution (k=3,G(7,5) ,8bit soft decision)	1/3
Inner code:Convolutional code(k=9,=1/2,G (753,561),8bit soft decision) Outer code: Proposed code (4Dm7, R=0.53)	1/4

Table 7-2. Simulation parameters.

Figure 7-7 shows the decoding circuit. The data received are fed to the memory cells of the torus-connected shift register. The connection order of the shift register is the same as the order of the encoder output shift register. The register contents are then transferred in parallel to the parity calculation circuit as well as the majority logic circuit to decide the digit. The majority logic circuit is a principal part of decoding, and whether or not the value exceeds the given threshold, each digit is corrected by a majority decision of the n independent parity checks. To improve the decoding characteristics, the decision of the majority logic circuit is repeated several times by varying the threshold value, and the digit value determined by the logic circuit is fed back to the input side register. After completion of decoding, only the data digit is transferred to the output shift register.

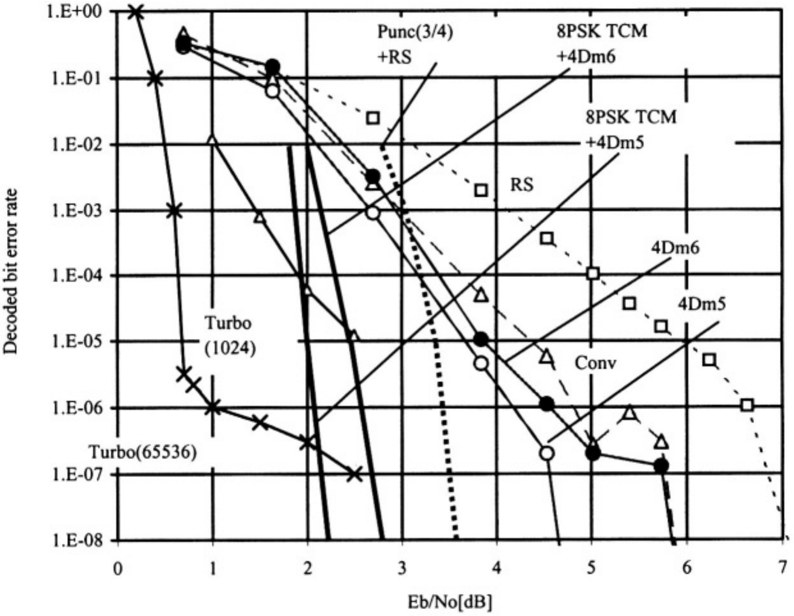


Figure 7-6. Improvement in decoded BER for random error by concatenation of the proposed code with convolution code with three memory cells.

Chip-Spec	
Circuit Size	100kGate NAND
Process	0.35 micron m
Chip Size	31.2*31.2mm
Package	QFP256Pin
Clock Speed	**4.1.1.1.1 66.6MHz**
Through-put	48Gbps
Input-Output	66.6*16M bps
Code-Performance	
Code	3-Dimension Size 9 (3Dm9)
Block Size	729 bit
Data Size	512 bit
Rate	0.70

Table 7-3. Outline of developed-chip for 3-dimension code

The outline of the LSI chip of the 3 dimensional size 9 code is shown in Table 7-3. The LSI chip consists of 100k gate of 0.35 micron, and has an input-output speed of 1 Gbps and a decoding throughput of 48Gbps. The LSI chips have been successfully applied to a trial radio transmission equipment of image of a Gbps class.

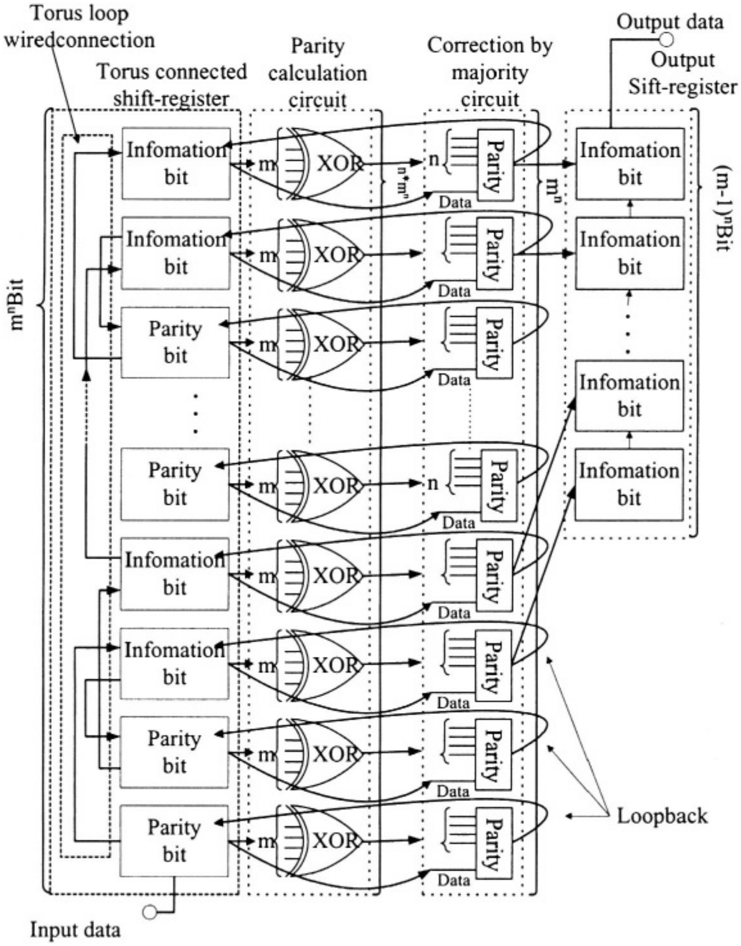


Figure 7-7. Decoder configuration

5. CONCLUSION

We proved that the hardware implementation of a torus knot code was effectively realized by using a unique circuit configuration, which maximizes the cyclic and symmetrical properties of the code. Furthermore, through performance simulations, the code was proven to have robust decoding characteristics with a degraded channel such as one hundredth to one tenth. In the near future, the high dimensional torus knot code is expected to find wide applications in the field of high-speed radio communications.

ACKNOWLEDGEMENTS

The work was supported by the '98 NEDO project of the Ministry of International Trade and Industry of Japan and by the 2000 IT-development support project for creating new industry for Terra-bit technology of the Ministry of Economy, Trade and Industry of Japan.

REFERENCES

[1]Hata M., Hamasuna Y., Yamaguchi E. and Takumi I., "High-speed and robust error correcting code for future mobile communications of high –dimensional discrete torus knot",WPMC'01, pp.367-372, Sept.2001, Aalborg, Denmark.
[2] Copyright Simon Rockliff, University of Adelaide, 1991. http://imailab-www.iis.u-tokyo.ac.jp/Members/robert-e.html
[3] Copyright Phil Karn,KA9Q,1995. http://imailab-www.iis.u-tokyo.ac.jp/Members/robert-e.html
[4] Copyright Yufei Wu, MPRG lab,Virginia Tech, 1998. http://www.ee.vt.edu/yufei/turbo.html
[5] Lin.K Y. and Lee J. "Results on the use of concatenated Reed-Solomon/Viterbi channel coding and data compression for space communications", *IEEE Trans.* Communications, Vol.COM-32, 5,pp. 18-523, May 1984.

Chapter 8

Modified Cerebellar Model Articulation Controller (MCMAC) As An Amplitude Spectral Estimator for Speech Enhancement

Abdul Wahab[1], Tan Eng Chong[1] , and Hüseyin Abut[2]

[1] *School of Computer Engineering, Nanyang Technological University Nanyang Avenue, Singapore;* [2]*Electrical and Computer Engineering Department, San Diego State University, San Diego, CA 92182, USA.* *Email: asabdul@ntu.edu.sg*

Abstract: In this chapter, we present a modified cerebellar model articulation controller (MCMAC) to be used together with the amplitude spectral estimator (ASE) for enhancing noisy speech. The MCMAC training overcomes the limitations of the CMAC technique we have employed noise/echo cancellation in a vehicular environment. While the CMAC in the training mode has trained only the trajectory it has visited by controlling the reference input, the modified MCMAC-ASE system architecture proposed in this work includes multiple MCMAC memory trainable for different noise sources.

Keywords: Cerebellar model articulation controller (CMAC), speech enhancement, echo cancellation, in-car noise, amplitude spectral estimation, Wiener filtering, Kohonen's self-organizing neural network (SFON), Grossberg learning rule, neighborhood function, and MOS.

1. INTRODUCTION AND CEREBELLAR MODEL ARITICULATION CONTROLLER

In this chapter, we present first a cerebellar model articulation controller (CMAC) block diagram as shown in Figure 8-1, which can be described as an associative memory that can be trained to implement non-linear functional mappings.

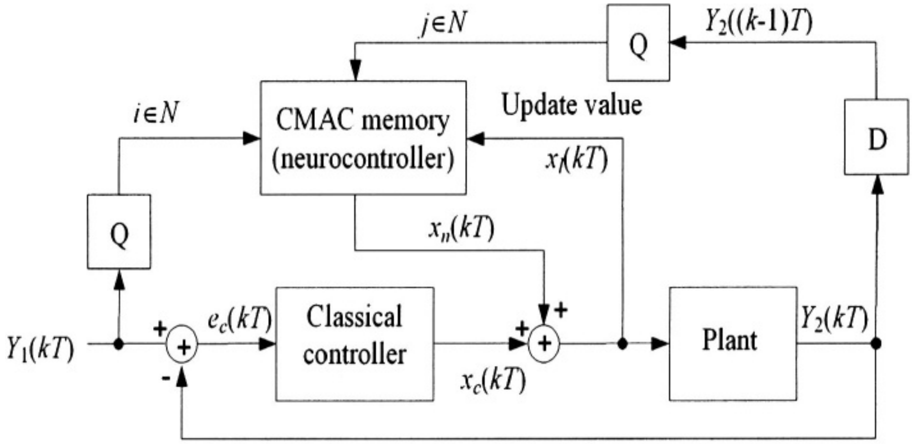


Figure 8-1. Cerebellar Model Articulation Controller (CMAC) Block Diagram.

The CMAC network in Figure 8-1 can be viewed as two layers of neurons, and hence, its operation can be decomposed into two separate mappings. The input vector is transformed to a vector of binary values, which, in turn, produces at the output the sum of weights that link itself to the corresponding input vector of value one. Given an input vector, the desired output at the output layer can be approximated by modifying its connection weights through the use of an adaptation process, commonly known as the training mode in the perceptron theory.

While there have been many studies in the area of CMAC memory and its associated architectures, most of them were concentrated in adaptive control problems and a simple two-dimensional CMAC configuration was sufficient for them.

In the case of signal processing applications, in particular, in speech and image processing problems because of the data sizes involved and the locally-stationary and locally ergodic nature of the underlying processes, there are a number of factors affecting the performance. These, in turn, require more sophisticated setups. In an earlier work, we have used the CMAC concept in speech enhancement and echo cancellation with encouraging results [6]. In particular, the CMAC was used in the Wiener filtering stage of an Amplitude Spectral Estimation (ASE) for noise and/or echo cancellation in moving vehicles.

A major challenge in that study was the need to train the system, which was unacceptably long. In this current modified configuration called MCMAC, we have attempted to overcome this drawback and yet to improve the performance even further.

It is difficult to train the CMAC memory because the characteristic surface has to be learned while a classical model controller controls the plant. For a particular control setting, the plant output typically follows a certain trajectory. Hence, only the weights of the output neurons (cells) visited by the path of this trajectory are updated, i.e., not every cell in the memory. This poses a major problem in many control situations that require on-line learning for which the control rules are not readily available.

We will now present the mathematical inner workings of the CMAC, which will be used later in the proposed MCMAC configuration. The CMAC memory of Figure 8-1 consists of a two-dimensional array that stores the value of the signal $x_n(kT)$ as the content of an element in the array with coordinates *i,j* [1]. The value at location at *i, j* is obtained by quantizing the reference input $y_{ref}(kT)$ and the plant output $y_p(kT)$. This quantization process can be described in the form of:

$$Q(y(kT)) = [\frac{n(y(kT) - y_{\min})}{y_{\max} - y_{\min}}] \tag{1}$$

where

$y_{\max}$ = Maximum value of *y(t)*,
$y_{\min}$ = Minimum value of *y(t)*, and
n = Resolution of the CMAC memory.

In addition, *k* represents a discrete step and T is the sampling period. During the initial operation, the plant receives almost all of its control input from the classical controller, while the CMAC memory is initialized to a set of preset values. During each subsequent control step, the classical control actuation signal $x_c(kT)$ is used for building a CMAC characteristic surface, which converges to the final surface at the end of the process.

The CMAC memory defined in previous few paragraphs can be visualized alternatively as a neural network consisting of a cluster of two-dimensional *self-organizing neural networks* (SOFM) in the field of expert systems. However, instead of a random initialization of the neural net weights, as it is normally done, here they are fixed such that they form a two dimensional neural grid as depicted in Figure 8-2.

The winning neuron in the CMAC memory at time step *k* is identified as the neuron with weights $Q(y_{ref}(kT))$ and $Q(y_p(kT\text{-}T))$, given the input values

$y_{ref}(kT)$ and $y_p(kT\text{-}T)$. The weights are effectively the coordinates (i, j) of the location of the neuron in the SOFM. The output of the winning neuron is obtained from the weight w_{ij} of the output neuron.

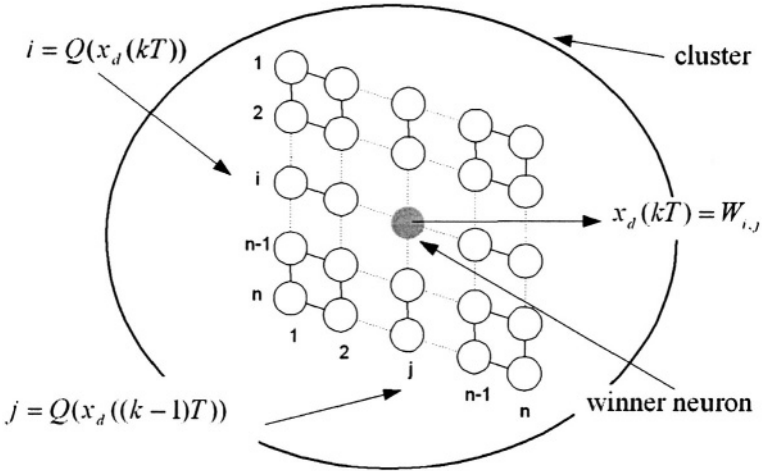


Figure 8-2. CMAC Memory Architecture.

As in the Kohonen's neural networks framework, the practice in CMAC learning is a competitive learning process and follows the well-known SOFM learning rules. With a simple caveat, since the weights of the cluster of neurons that represent indices to the CMAC memory are fixed, learning occurs only at the output neuron. To implement this, the learning rule for CMAC has been based on the well-known Grossberg competitive learning rule and it is applied only at the output layer. Furthermore, no competitive Kohonen learning rule is applied to the input layer. Therefore, the CMAC learning rule can be represented by [2]:

$$\begin{aligned} &i = Q(y_{ref}(kT)),\ j = Q(y_p(kT-T)); \quad i,j \in N \\ &w_{i,j}^{(k+1)} = w_{i,j}^{(k)} + \lambda(x(kT)) - w_{i,j}^{(k)}) \end{aligned} \tag{2}$$

where

λ = Learning parameter,

$x(kT)$ = Plant input at step k,

$y_{ref}(kT)$ = Reference input at step k,
$y_p(kT-T)$ = Plant output at step $k-1$,
$w_{i,j}^{(k)}$ = Contents of CMAC cell with coordinates *i,j* at step k, and
$Q(\circ)$ = Quantization function defined by equation (1).

2. MODIFIED CMAC (MCMAC) STRUCTURE

The *Modified CMAC* architecture, abbreviated by MCMAC, has been proposed [3] to overcome this problem by using the plant closed loop error $e_c(kT)$ and the plant output $y_p(kT)$ during the training. This allows on-line training as well as ease in the planning of the training trajectory. The training path can now be more directly controlled using the reference plant input $y_{ref}(kT)$. The proposed MCMAC block diagram is depicted in Figure 8-3.

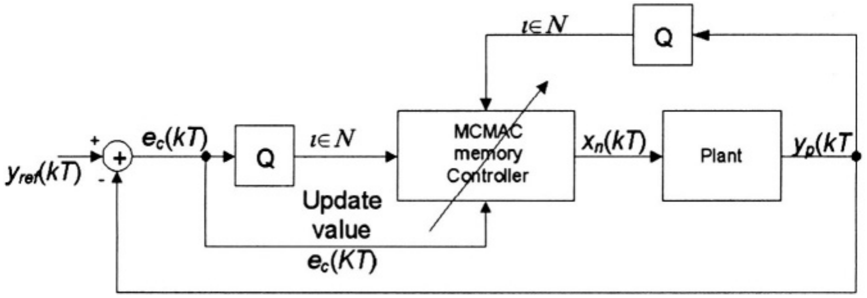


Figure 8-3. Proposed Modified Cerebellar Machine Articulation Controller (MCMAC).

The architecture of the proposed MCMAC is quite similar to a CMAC, except that the quantized closed loop error is employed, which is simply:

$$e_c(kT) = y_{ref}(kT) - y_p(kT)$$

This error signal and the plant output $y_p(kT)$ point to a two-dimensional MCMAC memory array instead of the original $y_{ref}(kT)$ and $y_p(kT-T)$ in the CMAC configuration. The learning rule for this modified architecture can be formulated by the following set of equations:

$$m = Q(e_c(kT)), n = Q(y_p(kT)); \quad m,n \in N \tag{3}$$

$$\begin{aligned} w_{m,n}^{(k+1)} &= w_{m,n}^{(k)} + \lambda e_c(kT) \\ &= w_{m,n}^{(k)} + h_{m,n}(\lambda(1-\alpha).e_c(kT) + \alpha \Delta w_{m,n}^{(k)}) \\ &\qquad for \, | m-i | \leq N, | n-j | \leq N; \quad i,j \in N \end{aligned} \tag{4}$$

$$h_{m,n} = e^{-| r_{m,n} - r_{i,j} |^2 / 2\sigma^2} \tag{5}$$

$$\Delta w_{m,n}^{(k+1)} = w_{m,n}^{(k+1)} - w_{m,n}^{(k)} \tag{6}$$

where

(m,n) = Cell coordinates,

kT = Sampling instant

λ = Learning parameter,

α = Momentum parameter,

$y_{ref}(kT)$ = Reference input at step *k,*

$y_p(kT)$ = Plant output at step *k,*

$e_c(kT) = y_{ref}(kT) - y_p(kT)$; closed-loop error,

N = Neighborhood parameter,

$| r_{m,n} - r_{i,j} |$ = Distance from cell located at (*m,n*) to cell *(i, j)*

σ^2 = Variance of Gaussian distribution,

$w_{m,n}^{(k)}$ = Contents of MCMAC cell located at *m, n* at step *k,* and

$Q(\circ)$ = Quantization function defined by equation (1).

It is worth noting that the original CMAC and the proposed MCMAC configurations attempt to model the characteristics of the plant on the basis of input indices. The difference between the two is that the former system learns from the plant input *x(kT)* and the output $y_p(kT)$, while the latter one employs the closed loop error $e_c(kT)$ and the plant output $y_p(kT)$ in their respective learning modes. Hence, the MCMAC does not require an inverse model of the plant and there is no need for the classical controller to be operational during the learning phase. This has an added advantage in the determination of the training trajectory through the control of the reference input $y_{ref}(kT)$.

Although this modification to CMAC has removed the need for a classical controller, the training of the modified cerebellar articulation controller (MCMAC) still requires a careful planning such that all of the cells in the MCMAC memory have to be visited. The contents of the MCMAC memory represent the plant characteristics to be controlled by the neuro-controller.

From our graph theory knowledge, the MCMAC memory can be also visualized as a 3-D characteristic surface, or the contour surface. The axis of

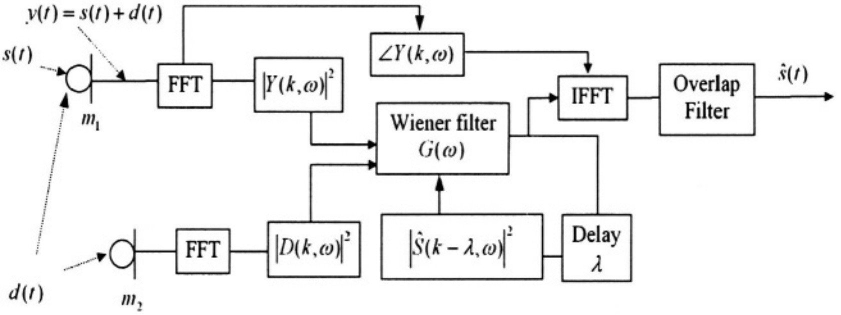


Figure 8-4. Noise Canceller-based on Amplitude Spectral Estimation and Wiener Filtering.

this contour surface consists of the cell indices *(m,n)* representing the locations with quantized values of the closed-loop error and the output $Q(e_c(kT))$ and $Q(y_p(kT))$, respectively, and the content of each cell. This information is subsequently used in the computation of the training rate for both the CMAC learning rule and the modified learning rule MCMAC.

In the framework of speech enhancement using the ubiquitous amplitude spectral estimation (ASE) techniques, we have employed the usual short-time Fourier transform method (STFT) to estimate the power spectral density for both the reference noise and the noisy speech [4, 5, 6]. To achieve that we have utilized a back-to-back configuration of a stereo microphone pair, as shown in Figure 8-4.

This block diagram and its numerous variations and extensions are very well-known in the speech processing community and we wanted to test our ideas in this framework. To retrofit the stereo microphone pair suitable to the vehicular systems we have placed them back-to-back as it has been regularly done in recording studios.

It is not difficult to see that the channel facing the speaker (m_1) is expected to contain mainly the speech from a speaker, whereas the opposite channel (m_2) will be primarily the reference signal (noise) and the secondary and ternary echoes reflected from the windshield and the back of the car, respectively. Understandably, there will be some portion of the reference signal in the front channel as well due to reflection from the chamber walls, the primary acoustic echo and the speech in the other one.

It was reported in a number of works in the automotive science that the corruption of speech in a vehicular environment is not purely additive in nature. In most cases, the relationship between the original speech and the noise involves a convolution process instead of a simple addition, which has been the norm in the Shannon-based information processing and communication systems community. In other words, the noisy speech can be better expressed by:

$$y(t) = s(t) * d(t) \tag{7}$$

Here $s(t)$ represents the speech input whereas, *d(t)* is the overall degradation, which may include the impulse response of the vehicular chamber. Since the model is not additive, Fourier analysis and the subsequent filtering cannot be applied directly. This, in turn eliminates the usage of the ubiquitous LMS-based ASE algorithm as it has been the case in the majority of earlier speech enhancement techniques including our earlier studies [4,6].

To overcome this, we have opted to resort to a high-order CMAC with a non-linear basis function. This has allowed us to tackle both the ambient convolutive noise term associated with the chamber and the traditional additive noise term a-la-communication systems.

Experimental results reported later in this chapter demonstrate not only the effectiveness of the proposed MCMAC system when coupled with an ASE in the enhancement of the convolutive nature of noise but also the robustness of the technique promises as a viable candidate for deployment in the next generation vehicular communication systems.

The Signal plus Noise to Noise Ratio (SNNR) has been used as the quantitative measure of performance in this work and it can be defined as the sum of the *a priori* SNR and the *a posteriori* SNR values:

$$SNR_{Total} = (1-\beta)SNR_{posteriori} + \beta.SNR_{priori} \tag{8}$$

where $\beta \in (0,1)$. It is not difficult to conclude from Figure 8-4 that the SNR_{priori} and the $SNR_{posteriori}$ are given by:

$$SNR_{posteriori} = \left| \frac{|Y_k(\omega)|^2}{E\{|D(w)|^2\}} \right| - 1 \tag{9a}$$

$$SNR_{priori} = \frac{|H_{k-\lambda}(\omega)Y_{k-\lambda}(\omega)|^2}{E\{|D(\omega)|^2\}} = \frac{|\hat{S}_{k-\lambda}(\omega)|^2}{E\{|D(\omega)|^2\}} \tag{9b}$$

In the training mode, Equations (9a,9b) and their simplified approximations have been used to obtain the weights for the Weiner filter. These weights were then stored into the MCMAC memory.

In the recall mode, however, all the memory elements, pointed by the SNR_{priori} and the $SNR_{posteriori}$ as address indices, are added together with respect to a neighborhood basis function. A neighborhood basis function is needed to restrict the impact of various memory locations in the computation of the final winning neuron.

The neighborhood function $f(x)$ can be a simple average or an algorithm based on a uniform distribution of errors in a specific region of memory weights. In practice, spline functions have been the most popular basis functions for higher-order CMAC systems.

In our experiments, we have chosen a Gaussian neighborhood function as the basis function:

$$f(x) = e^{-\frac{1}{2}(\frac{a_{i,j} - x_{i,j}}{\sigma})^2} \tag{10}$$

where σ^2 is the variance of a Gaussian distribution and $a_{i,j} - x_{i,j}$ is the distance between a cell with coordinates *(i,j)* and the input with the same coordinates. The details of the learning rule and the associated neighborhood function and their implementations can be found in [7].

3. ASE-MCMAC ALGORITHM

In Figure 8-5, we propose a speech enhancement system including an MCMAC where the Weiner filter coefficients are constantly updated. The variance of the signal to noise ratio is used as the address index to a specific MCMAC memory location and the third dimension of the MCMAC memory is the frequency.

The ASE algorithm as configured in Figure 8-5 operates in the frequency-domain since we need to compute the spectral value used in SNR computation (8-9a, 9b). Using the learning rule as expressed in (3-6), we compute the profile of the Weiner filter weights and store them in the MCMAC memory. These new or updated values from the MCMAC memory are read and used as the Weiner filter weights in estimating the enhanced speech without any musical noise artefacts. Subsequently, we use the enhanced speech to calculate the SNR values needed in the ASE module and the SNR variance module. Therefore, this closed-loop nature of the ASE-MCMAC allows the system to employ an unsupervised learning.

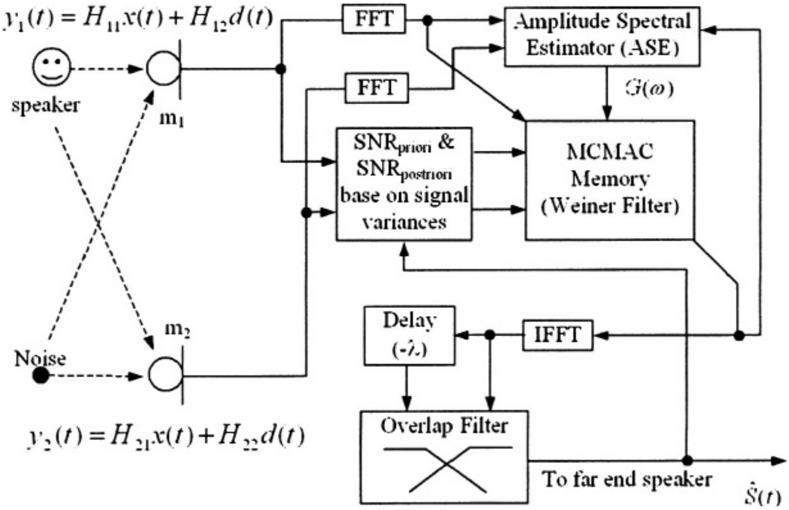


Figure 8-5. Block Diagram of the Proposed ASE-MCMAC Using a Microphone Pair.

In the recall mode, however, the information needed for the computation of address indices are all in the time-domain, except for the processing of the noisy signal by the Wiener filter. We could have carried out this step in the frequency-domain as well, but that would have required a number of FFT modules. As expected, working partly in the frequency-domain and partly in the time-domain the overall processing time has been shortened considerably.

Along the lines of many practical echo canceller systems reported in literature, we have performed a frame-based updating strategy for our Wiener filter coefficients as opposed to the sample-by-sample computations. In addition to yielding considerable computational savings, this frame-by-frame approach turns out to be consistent with the subsequent speech compression stage, where one of the existing standards is normally employed including LPC-10, RTP, CELP, MELP, and their derivatives.

4. EXPERIMENTS AND DISCUSSION OF RESULTS

To evaluate the performance of the proposed technique the ASE-MCMAC noise cancellation system of Figure 8-5, has been built as well as the benchmark LMS-ASE system. The microphones m_1 and m_2 were presented with a pre-recorded speech and noise data collected in a moving vehicle subject to varying environmental conditions. We will refer these microphones as the primary channel microphone m_1, which has been placed towards the speaker and the noise reference microphone m_2, which was faced away from the speaker, respectively. It is worth noting that this back-to-back placement of microphones is the standard procedure in recoding studios. Using several combinations of SNR values for m_1 and $m_{2,}$ we have created noisy speech, i.e., low-noise, medium-noise, and high-noise regimes.

In Figure 8-6, we illustrate the performance of our speech denoising experiments based on the proposed MCMAC-ASE algorithm against the benchmark LMS-ASE algorithm under various noise conditions.

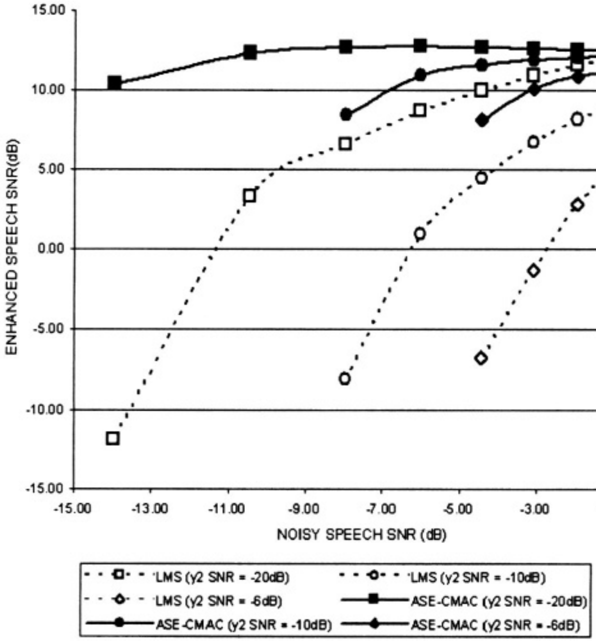


Figure 8-6. Performance of ASE-MCMAC and LMS under Different Noise Regimes.

At low noise (high SNR) cases, the improvement over the benchmark is insignificant, i.e., any technique would work as expected.

However, at high noise (low SNR) regimes, the case for a vehicle moving in heavy traffic, the LMS algorithm clearly fails. On the other hand, the ASE-MCMAC algorithm has performed remarkably well. We would like to note that this conclusion is valid as long as the SNR of the noise reference input into the microphone m_2 is less than that of the primary channel microphone m_1.

The advantage of the ASE-MCMAC over the classical ASE approach is that once the noise spectrum has been trained only a recall process is needed for processing incoming speech. Even though fairly encouraging results were obtained in our earlier work based on a CMAC, the performance of the ASE based on this modified version (MCMAC-ASE) is remarkably superior to its predecessor. Response from quite a number of experienced listeners has been uniformly the same.

In addition during training, the corrupt speech is also processed as well. Recalling the fact that only a subset of the overall memory of the CMAC were trained while for the MCMAC case, almost all of the memory locations in a fairly large neighborhood were trained during the learning cycle.

To gain more inside about the implications of the remarkable results from MCMAC experiments, we plot the Wiener filter profiles for both the classical CMAC and the current MCMAC algorithms in Figures 8-7.a and 8-7.b, respectively. Thus, no further training is required for a fully trained MCMAC memory unless the profile of the Weiner filters changes drastically. The MCMAC memory once fully trained can be used without any further training. Multiple of this MCMAC memory may be required from different training noise environment to form a complete system.

Nevertheless we would like to caution the reader that the ultimate judgment would come from the formal MOS -based evaluations performed under controlled stimuli. We believe the success of the proposed method could be attributed to the fact that only a subset of the CMAC memory was trained. For the case of MCMAC, however, almost all of the memory locations in a fairly large neighborhood have been trained during the learning cycle.

5. CONCLUSIONS

In this work we have presented a modified cerebellar model articulation controller (MCMAC) and its deployment in the vehicular environment as an integral part of the speech enhancement system. Even though, the performance of the ASE-MCMAC algorithm is comparable to that of a traditional LMS-based noise cancellation algorithm high SNR values greater than –2dB, but for low SNR (high noise) regimes the proposed ASE-MCMAC performs remarkably well and it has been shown to be very robust in different type of noise in vehicles including the engine noise, road noise, tire tractions, the wind and rain.

Provided sufficient computing power is available multiple MCMAC memories can be employed to get additional gains as it is recently done in microphone array-based implementations. In a recent report, this idea has been explored with encouraging results within the framework of environmental sniffing [8].

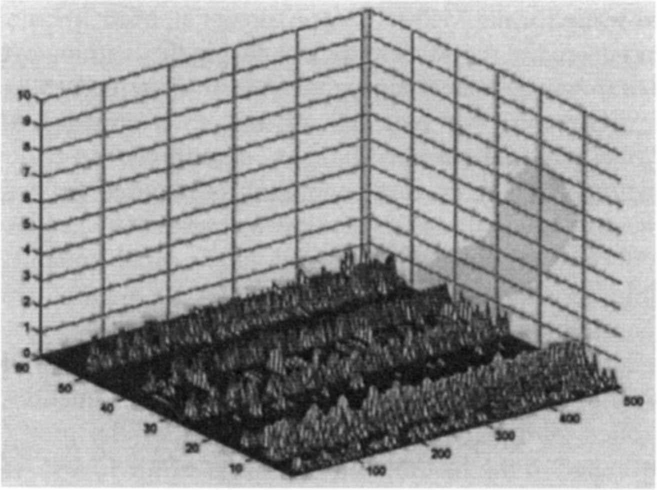


Figure 8-7.a. CMAC Profile for the ASE Wiener Filter.

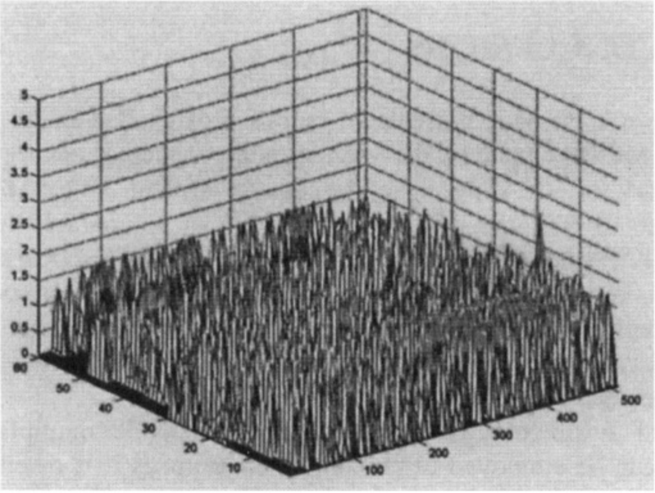


Figure 8-7.b. MCMAC Profile for the ASE Wiener Filter.

REFERENCES

[1] Kraft, L.G., and Campagna, D. P., A Comparison of CMAC Neural Network Control and Two Traditional Adaptive Control Systems, IEEE Control Systems Magazine, 36, 1990.

[2] Zurada J. M., Introduction to Artificial Neural Systems, Info Access Distribution Pte. Ltd., Singapore, 1992.

[3] C.Quek and P.W.Ng., Realisation of Neural Network Controllers in Integrated Process Supervision, Inter. Journal of Artificial Intelligence in Engineering, 10(2), 135, 1996

[4] Ephraim, Y., and Malah, D., Speech enhancement using minimum mean square error short-time spectral amplitude estimator. IEEE Transaction on Acoustics, Speech, and Signal Processing, ASSP-32, 6, 1109, 1984.

[5] Jeannès, R. Le B., Faucon, G. and Ayad, B., How to Improve Acoustic Echo and Noise Cancelling using a Single Talk Detector. *Speech Communication,* 20, 191, 1996.

[6] Abdul, W., Tan, E. C, and Abut, H., Robust Speech Enhancement Using Amplitude Spectral Estimator, *Proceedings of the IEEE ICASSP2000 Silver Anniversary,* Vol. VI, 3558, 2000.

[7] Abdul W, "Speech Enhancement in Vehicular Environment." Unpublished Ph.D. Thesis, Nanyang Technological University, Singapore, 2003.

[8] M. Akbacak, and J. H. L. Hansen, "Environmental Sniffing: Noise Knowledge Estimation for Robust Speech Systems," Proceedings IEEE ICASSP2003, Vol. 2, pp. 113-116, Hong Kong, April 2003.

This page intentionally left blank

Chapter 9

NOISE ROBUST SPEECH RECOGNITION USING PROSODIC INFORMATION

Koji Iwano, Takahiro Seki, Sadaoki Furui

Department of Computer Science, Tokyo Institute of Technology 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552 Japan Email: iwano@furui.cs.titech.ac.jp

Abstract This paper proposes a noise robust speech recognition method for Japanese utterances using prosodic information. In Japanese, the fundamental frequency (F_0) contour conveys phrase intonation and word accent information. Consequently, it also conveys information about prosodic phrase and word boundaries. This paper first proposes a noise robust F_0 extraction method using the Hough transform, which achieves high extraction accuracy under various noise environments. Then it proposes a robust speech recognition method using syllable HMMs which model both segmental spectral features and F_0 contours. We use two prosodic features combined with ordinary cepstral parameters: a derivative of the time function of $\log F_0$ ($\Delta \log F_0$) and a maximum accumulated voting value of the Hough transform representing a measure of F_0 continuity. Speaker-independent experiments were conducted using connected digits uttered by 11 male speakers in various kinds of noise and SNR conditions. It was confirmed that both prosodic features improve the recognition accuracy in all noise conditions, and the effects are additive. When using both prosodic features, the best absolute improvement of digit accuracy is about 4.5%. This improvement was achieved by improving the digit boundary detection by using the robust prosodic information.

Keywords: noise robust speech recognition, prosody, fundamental frequency (F_0), Hough transform

1. INTRODUCTION

How to increase robustness is one of the most important issues in building speech recognition systems in mobile and vehicular environments.

It has been found that human beings use prosodic information to increase the robustness in recognizing speech when acoustic information is unreliable [1]. Since the fundamental frequency (F_0) contour is one of the most important features for conveying Japanese prosody, it is expected to be useful for increasing the robustness of automatic speech recognition. F_0 contour information has already been used for improving the preformance of Japanese phoneme recognition in clean condition[2]. However, with the present technology, it is not easy to automatically extract correct F_0 values, especially in noisy environments. Various techniques have been proposed to smooth out incorrect values from a time series of extracted F_0 values, but these methods are not always successful. This paper proposes a novel robust method, in which the Hough transform is applied to a windowed time series of cepstral vectors extracted from speech, instead of directly extracting F_0 independently for each frame of speech. Due to its capability of extracting straight-line components from an image, the Hough transform can extract a reliable F_0 value for each window. By shifting the window at every frame, a smooth time function of F_0 can be obtained.

We also propose a speech recognition method using prosodic features extracted by the Hough transform, consisting of a derivative of the time function of $\log F_0$ ($\Delta \log F_0$) or/and a measure of periodicity. These features are combined with ordinary cepstral parameters and modeled by multi-stream HMMs, which are trained using clean speech. Since F_0 contours represent phrase intonation and word accent in Japanese utterances, prosodic features are useful to detect prosodic phrases and word boundaries. Therefore, the proposed method using robust prosodic information is able to precisely detect word boundaries and improve recognition performance under noisy environments.

The paper is organized as follows. In Section 2, a robust F_0 extraction method using the Hough transform is proposed. Section 3 describes our modeling scheme for noise robust speech recognition using syllable HMMs combining segmental and prosodic information. Experimental results are reported in Section 4, and Section 5 concludes this paper.

2. F_0 EXTRACTION USING THE HOUGH TRASNFORM

2.1 Hough Transform

The Hough transform is a technique to robustly extract parametric patterns, such as lines, circles, and ellipses, from a noisy image[3].

The Hough transform method to extract a significant line from an image on the x–y plane can be formulated as follows. Suppose the image consists of n pixels at (x_i, y_i) $(i = 1, \cdots, n)$. Every pixel on the x–y plane is transformed to a line on the m–c plane as

$$c = -x_i m + y_i \quad (i = 1, \cdots, n) \tag{9.1}$$

A brightness value of the pixel on the x–y plane is accumulated at every point on the line. This process is called "voting" to the m–c plane. After voting for all the pixels, the maximum accumulated voting value on the m–c plane is detected, and the peak point (m, c) is transformed to a line on the x–y plane by the following equation:

$$y = mx + c \tag{9.2}$$

2.2 F_0 Extraction Using the Hough Transform

Cepstral peaks extracted independently for each short period of speech have been widely used to extract F_0 values. This method often causes errors, including half pitch, double pitch and drop outs, for noisy speech. Since F_0 contours have temporal continuity in voiced periods, the Hough transform, taking advantage of its continuity, applied to time-cepstrum images is expected to have robustness in extracting pitch in the noisy environment.

Speech waveforms are sampled at 16kHz and transformed to 256 dimensional cepstra. A 32ms-long Hamming window is used to extract frames every 10ms. For reducing noise effects of a high frequency domain, we extract and use time-cepstrum images which are limited to 60~256 dimensions and liftered according to the following formula:

$$c'_d = \left\{0.6 + 0.4 \sin\left(\frac{d - 60}{140 - 60} \times \frac{\pi}{2}\right)\right\} \cdot c_d \tag{9.3}$$

where c_d is the original dth cepstrum and c'_d is the liftered cepstrum.

To the liftered time-cepstrum image, a nine-frame moving window is applied at every frame interval to extract an image for line information detection. The time-cepstrum image is used as the pixel brightness image for the Hough transform. An F_0 value is obtained from a cepstrum index of the center point for the detected line. Since the moving window has nine frames, the time continuity for 90ms is taken into account in this method.

In conventional F_0 extraction methods, F_0 values are extracted independently at every frame and various smoothing techniques are applied afterwards. The problem of these methods is that they are sensitive to a decrease in correctness of the raw F_0 values. Since our method uses the continuity of cepstral images, it is expected to be more robust than conventional methods.

2.3 Evaluation of F_0 Extraction

Utterances from two speakers, one male and one female, were selected from the ATR continuous speech corpus to evaluate the proposed method. Each speaker uttered 50 sentences. This corpus has correct F_0 labels given manually. White noise, in-car noise, exhibition-hall noise, and elevator-hall noise were added to these utterances at three SNR levels: 5, 10, and 20dB. Accordingly, 1,200 utterances were made for evaluation.

The correct F_0 extraction rate was defined as the ratio of the number of frames in which extracted values were within ±5% from the correct F_0 values to the total number of labeled voice frames.

Evaluation results showed that the extraction rate averaged over all noise conditions was improved by 11.2% in absolute value from 63.6% to 74.8%, compared to the conventional method without smoothing.

3. INTEGRATION OF SEGMENTAL AND PROSODIC INFORMATION FOR NOISE ROBUST SPEECH RECOGNITION

3.1 Japanese Connected Digit Speech

The effectiveness of the F_0 information extracted by the proposed method on speech recognition was evaluated in a Japanese connected digit speech recognition task. In Japanese connected digit speech, two or three digits often make one prosodic phrase. Figure 9-1 shows an example of the F_0 contour of connected digit speech. The first two digits make the first prosodic phrase, and the latter three digits make the second prosodic phrase. The transition of F_0 is represented by CV syllabic units, and each CV syllable can be prosodically labeled as a "rising", "falling", or "flat" F_0 part. Since this F_0 feature changes at digit boundaries, the accuracy of digit alignment in the recognition process is expected to be improved by using this information.

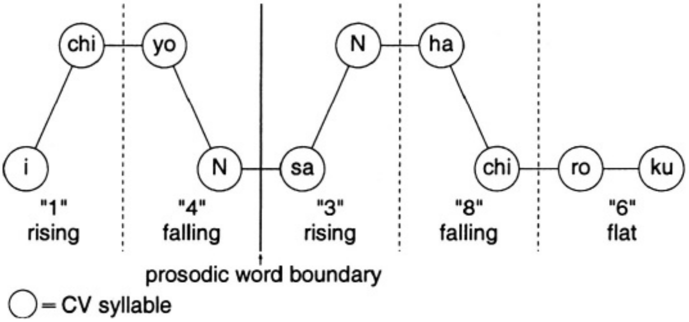


Figure 9-1. An example of F_0 contour of Japanese connected digit speech.

3.2 Integration of Segmental and Prosodic Features

Each segmental feature vector has 25 elements consisting of 12 MFCC, their deltas, and the delta log energy. The window length is 25ms and the frame interval is 10ms. Cepstral mean subtraction (CMS) is applied to each utterance.

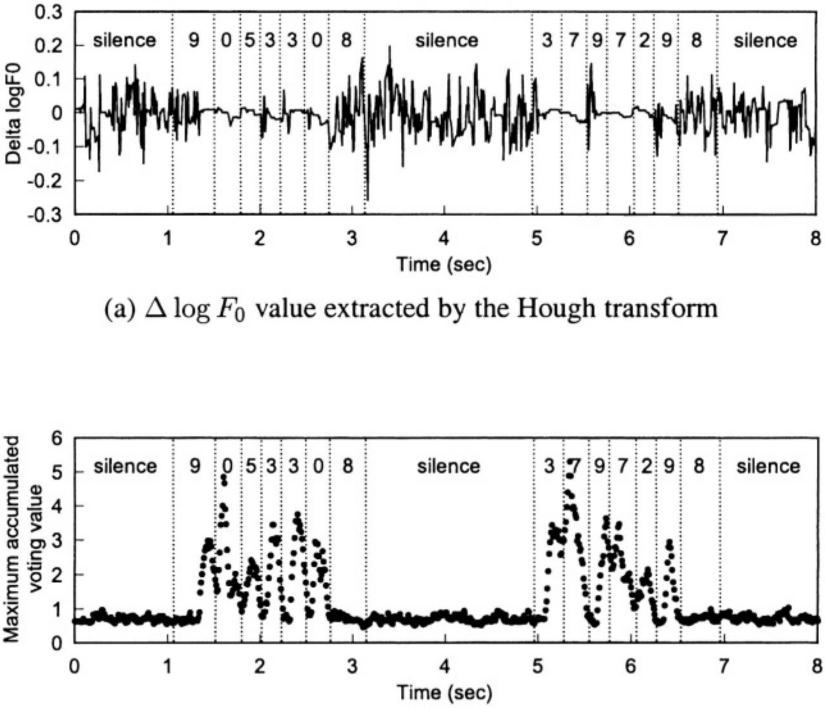
Two prosodic features are computed: one is the $\Delta \log F_0$ value which represents the F_0 transition, and the other is the maximum accumulated voting value obtained in the Hough transform which indicates the degree of temporal continuity in the F_0.

$\Delta \log F_0$ value is calculated as follows:

$$\begin{aligned}\Delta \log F_0 &= \frac{d \log F_0}{dt} \\ &= \frac{d \log F_0}{dF_0} \cdot \frac{dF_0}{dt} \\ &= \frac{1}{F_0} \cdot \Delta F_0 \end{aligned} \qquad (9.4)$$

ΔF_0 is directly computed from the line extracted by the Hough transform.

An example of the time function of the $\Delta \log F_0$ and maximum accumulated voting values is shown in Figure 9-2. A male speaker's utterance, "9053308" "3797298", with white noise added at 20dB SNR is shown. In unvoiced and pause periods, the $\Delta \log F_0$ fluctuates more than in voiced periods. The maximum accumulated voting values in unvoiced and pause periods are much smaller than that in voiced periods. These features are expected to be effective for detecting boundaries between voiced and unvoiced/pause periods.



(a) $\Delta \log F_0$ value extracted by the Hough transform

(b) Maximum accumulated voting value

Figure 9-2. An example of the prosodic features in Japanese connected digit speech for a male speaker's utterance, "9053308" "3797298", with 20dB SNR white noise.

In this paper, two kinds of prosodic features and their combination, **P-D, P-V,** and **P-DV,** are investigated:

P-D: $\Delta \log F_0$

P-V: maximum accumulated voting value

P-DV: $\Delta \log F_0$ + maximum accumulated voting value

These three kinds of prosodic features are combined with segmental features for each frame. Therefore, three kinds of segmental-prosodic feature vectors are built and evaluated.

3.3 Multi-stream Syllable HMMs

3.3.1 Basic Structure of Syllable HMMs. Since CV syllable transition and the change of F_0 characteristics such as "rising", "falling" and "flat" are highly related, the segmental and prosodic features are integrated using syllabic unit HMMs. Our preliminarily experiments showed that the syllable unit HMMs have approximately the same digit recognition accuracy for a connected digit task as tied-state triphone HMMs.

The integrated syllable HMM denoted by "SP-HMM (Segmental-Prosodic HMM)" models both phonetic context and F_0 transition. Table 9.1 is the list of SP-HMMs used in our experiments. Each Japanese digit uttered continuously with other digits can be modeled by a concatenation of two context-dependent syllables. Even "2" (/ni/) and "5" (/go/) can be modeled by two syllables since their final vowel is often lengthened as /ni:/ and /go:/. The context of each syllable is considered only within each digit in our experiment. Therefore, each SP-HMM is denoted by either a left-context dependent syllable "LC-SYL, PM" or a right-context dependent syllable "SYL+RC, PM", where "PM" indicates a F_0 transition pattern which is either rising ("U"), falling("D") or flat("F"). For example, "the first syllable /i/ of "1" (/ichi/) which has rising F_0 transition" is denoted as "i+chi,U". Each SP-HMM has a standard left-to-right topology with $n \times 3$ states, where n is the number of phonemes in the syllable. "sil" and "sp" models are used for representing a silence between digit strings and a short pause between digits, respectively.

3.3.2 Multi-stream Modeling. SP-HMMs are modeled as multi-stream HMMs. In the recognition stage, the probability $b_j(O_{SP})$ of generating segmental-prosodic observation O_{SP} at state j is calculated by:

$$b_j(O_{SP}) = b_j(O_S)^{\lambda_S} \cdot b_j(O_P)^{\lambda_P} \tag{9.5}$$

where $b_j(O_S)$ is the probability of generating segmental features O_S and $b_j(O_P)$ is the probability of generating prosodic features O_P. λ_S and λ_P are weighting factors for the segmental and prosodic streams, respectively. They are constrained by $\lambda_S + \lambda_P = 1$.

3.3.3 Building SP-HMMs. Syllable HMMs for segmental and prosodic features are separately made and combined to build SP-HMMs using a tied-mixture technique as follows:

1 "S-HMMs (Segmental HMMs)" are trained by using only segmental features. They are denoted by either "LC-SYL,*" or "SYL+RC,*". Here, "*"

digit	model			digit	model		
0	ze+ro,U	ze+ro,D	ze+ro,F	6	ro+ku,U	ro+ku,D	ro+ku,F
/zero/	ze-ro,U	ze-ro,D	ze-ro,F	/roku/	ro-ku,U	ro-ku,D	ro-ku,F
1	i+chi,U	i+chi,D	i+chi,F	7	na+na,U	na+na,D	na+na,F
/ichi/	i-chi,U	i-chi,D	i-chi,F	/nana/	na-na,U	na-na,D	na-na,F
2	ni+i,U	ni+i,D	ni+i,F	8	ha+chi,U	ha+chi,D	ha+chi,F
/ni:/	ni-i,U	ni-i,D	ni-i,F	/hachi/	ha-chi,U	ha-chi,D	ha-chi,F
3	sa+N,U	sa+N,D	sa+N,F	9	kyu+u,U	kyu+u,D	kyu+u,F
/saN/	sa-N,U	sa-N,D	sa-N,F	/kyu:/	kyu-u,U	kyu-u,D	kyu-u,F
4	yo+N,U	yo+N,D	yo+N,F		sil	sp	
/yoN/	yo-N,U	yo-N,D	yo-N,F				
5	go+o,U	go+o,D	go+o,F				
/go:/	go-o,U	go-o,D	go-o,F				

Table 9-1. List of SP-HMMs (Segmental-Prosodic HMMs). SP-HMM is denoted by either “LC-SYL,PM” or “SYL+RC,PM”. “LC-SYL” indicates the left-context dependent syllable and “SYL+RC” indicates the right-context dependent syllable. “PM” indicates F_0 pattern which is either rising(“U”), falling(“D”), or flat(“F”).

(wild card) means that HMMs are built without considering the F_0 transitions, “U”, “D” or “F”. The total number of S-HMM states is the same as the number of SP-HMM states. Twenty S-HMMs including “sil”, “sp” are trained.

2 Training utterances are segmented into syllables by the forced-alignment technique using the S-HMMs; and then, one of the F_0 transition labels, “U”, “D” or “F”, is manually given to each segment according to its actual F_0 pattern.

3 “P-HMMs (Prosodic HMMs)”, having a single state, are trained by prosodic features within these segments, according to the F_0 transition label. Eight separate models, “*-*,U”, “*+*,U”, “*-*,D”, “*+*,D”, “*-*,F”, “*+*,F”, “sil” and “sp”, are made. Each P-HMM has a single state, since it has been found that syllabic F_0 contours in Japanese can be approximated by a line function[4] and that the $\Delta \log F_0$ value can be expected to be almost constant in each CV syllable.

4 The S-HMMs and P-HMMs are combined to make SP-HMMs. Gaussian mixtures for the segmental feature stream of SP-HMMs are tied with corresponding S-HMM mixtures, while the mixtures for the prosodic feature stream are tied with corresponding P-HMM mixtures. Figure 9-3

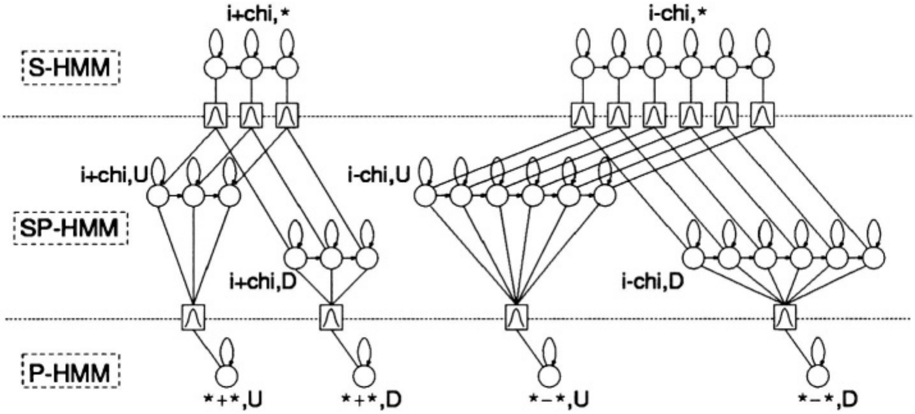


Figure 9-3. Building SP-HMMs using a tied-mixture technique. S-HMMs and P-HMMs are trained using segmental and prosodic features, respectively.

shows the integration process. In this example, mixtures of SP-HMM "`i+chi,U`" are tied with those of S-HMM "`i+chi,*`" and P-HMM "`*+*,U`".

4. EXPERIMENTS

4.1 Database

A speech database was collected from 11 male speakers in a clean/ quiet condition. The database comprised utterances of 2-8 connected digits with an average of 5 digits. Each speaker uttered the digit strings, separating each string with a silence period. 210 connected digits and approximately 229 silence periods were collected per speaker.

Experiments were conducted using the leave-one-out method; data from one speaker were used for testing while data from all other speakers were used for training, and this process was rotated for each speaker. Accordingly, 11 speaker-independent experiments were conducted, and a mean word accuracy was calculated as the measure of recognition performance. All the HMMs were trained using only clean utterances, and testing data were contaminated with either white, in-car, exhibition-hall, or elevator-hall noise at three SNR levels: 5, 10 and 20dB.

4.2 Dictionary and Grammar

In the recognition dictionary, each digit had three variations considering the F_0 transitions. For instance, variations of "1" comprised "`i+chi,U i-chi,U sp`", "`i+chi,D i-chi,D sp`", and "`i+chi,F i-chi,F sp`". This means that the F_0 transition pattern was not allowed to change within each digit. The recognition grammar was created so that all digits could be connected without any restrictions.

4.3 Experimental Results

Training and testing were performed using the HTK[5]. In our preliminary experiments, the best S-HMM recognition performance ("baseline") was obtained when the number of mixtures in each S-HMM was four. Experiments for selecting the optimum number of mixtures for the prosodic stream (P-HMMs) in SP-HMMs tied with four mixture S-HMMs were conducted, and the best performance using SP-HMMs was obtained when four mixture P-HMMs were used. Therefore, in the experiments hereafter, SP-HMMs were tied with four mixture S-HMMs and four mixture P-HMMs.

Table 9-2 shows the digit accuracy using SP-HMMs in various SNR conditions. **"SP-HMM-X"** indicates the SP-HMMs using the prosodic feature "**P-X**". Accuracies for four kinds of noises are averaged at 20, 10, and 5dB SNR, respectively. The segmental and prosodic stream weights and insertion penalties

SNR	S-HMM (baseline)	SP-HMM-D	SP-HMM-V	SP-HMM-DV
clean	99.3	99.6	99.4	99.4
20 dB	84.9	86.0	85.7	86.1
10 dB	53.1	54.6	55.1	55.7
5 dB	40.1	41.4	42.2	42.7

Table 9-2. Digit recognition accuracies by SP-HMMs and S-HMMs in various SNR conditions.

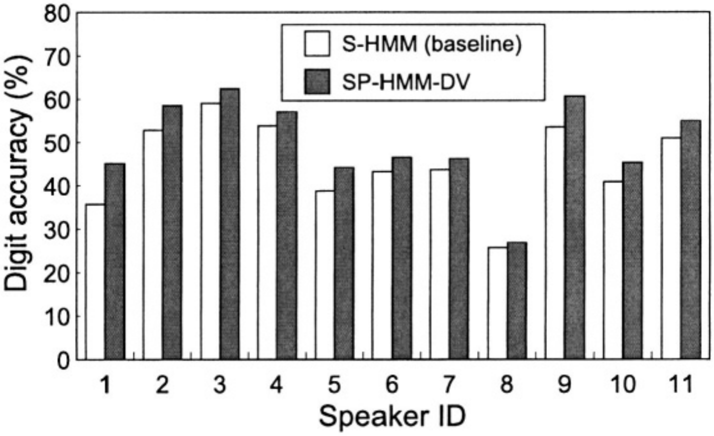


Figure 9-4. Comparison of the digit error rates by **SP-HMM and S-HMM-DV** for each speaker. In this experiment, 10dB exhibition-hall noise was added to the test set.

were optimized for each noise condition. Digit accuracies were improved in all kinds of noise and prosodic feature conditions. It can be seen that **SP-HMM-DV** showed the best performance, which means that the effects of the $\Delta \log F_0$ and the maximum accumulated voting value are additive. The best improvement of 4.5% from 45.3% to 49.8% is observed in the condition when exhibition-hall noise was added at 10dB SNR and the prosodic feature **P-DV** was used.

In Figure 9-4, the digit recognition accuracies by **S-HMM** and **SP-HMM-DV** are shown for each speaker. In this experiment, 10dB exhibition-hall noise was added to the test set. The improvement was observed for every speaker, which means that the proposed method is useful for speaker-independent recognition.

Figure 9-5 shows the improvement of digit recognition accuracy as a function of the prosodic stream weight λ_P at each SNR. Results for four kinds of noises

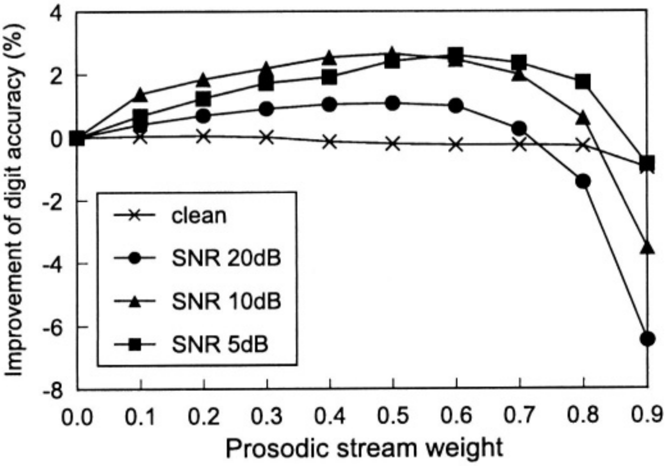


Figure 9-5. Improvement of digit accuracy as a function of prosodic stream weight (λ_P) in each SNR condition.

are averaged at 20, 10, and 5dB SNR, respectively. In this experiment, the prosodic feature **P-DV** was used, and insertion penalties were optimized. The improvement using the SP-HMMs was observed over a wide range: $0.0 < \lambda_P \leq 0.7$ in all the noise conditions. Best results were obtained when λ_P was set aroung 0.6, irrespective of the SNR level.

Figure 9-6 shows the optimum insertion penalty as a function of the prosodic stream weight λ_P in the white noise condition, when the prosodic feature **P-DV** was used. In noisy conditions, if the prosodic stream weight is low, we need to set the insertion penalty high to compensate for the low reliability of segmental features. Since prosodic features are effective for digit boundary detection, the higher the prosodic stream weight becomes, the lower the optimum insertion penalty becomes. Similar results were obtained for other noise conditions. The control range of the optimum insertion penalties in the best prosodic stream weight condition ($\lambda_P = 0.6$) is approximately a half of the range for the condition without using the prosodic information. This means that the prosodic features are effective for robust adjustment of the insertion penalty.

As a supplementary experiment, we compared the boundary detection capability of SP-HMMs and S-HMMs for digit recognition under noisy environments. Noise-added utterances and clean utterances were segmented by both of these models using the forced-alignment technique. The boundary detection

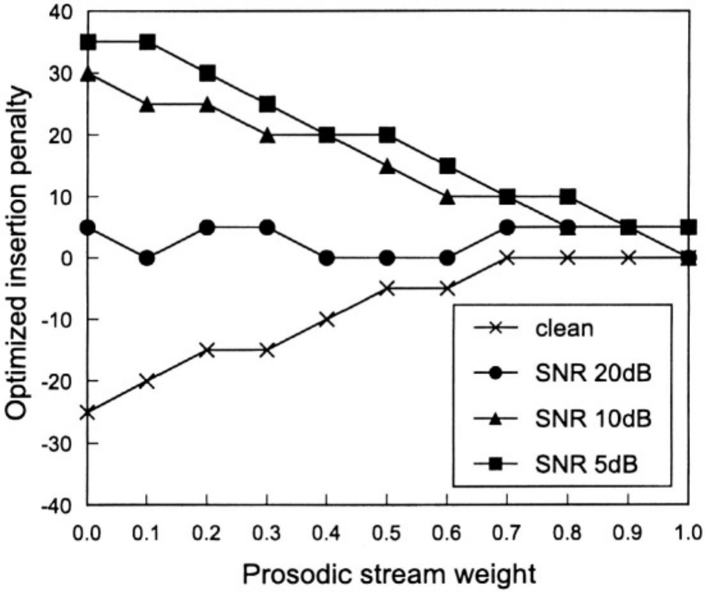


Figure 9-6. Optimized insertion penalty as a function of prosodic stream weight (λ_P) in white noise condition.

errors (ms) were measured by comparing the detected boundary locations in noise-added utterances with that in clean utterances. The mean digit boundary detection error rate was reduced by 23.2% for 10dB SNR utterances and 52.2% for 5dB SNR utterances using the **SP-HMM-DV.** These results indicate the effectiveness of prosodic information in digit boundary detection.

5. CONCLUSIONS

This paper has proposed an F_0 extraction method using the Hough transform and a new speech recognition method using syllable HMMs utilizing both segmental and prosodic information. Both methods were confirmed to be robust in various noise conditions. The prosodic information is effective in digit boundary detection and consequently improves connected digit recognition performance under noise. Future works include combination of our method with model adaptation or feature normalization techniques for noise effects and evaluation using more general recognition tasks.

References

[1] Kori, S. (1996). “Onsei no tokucho kara mita bun,” Nihongogaku, vol.15, no.8, pp.60–70. (in Japanese)

[2] Iwano, K. and Hirose, K. (1999). “Prosodic word boundary detection using statistical modeling of moraic fundamental frequency contours and its use for continuous speech recognition,” Proc. Inter. Conf. on Acoustics, Speech and Signal Proc., Phoenix, Arizona, vol.1, pp. 133–136.

[3] Hough, P.V.C. (1962). “Method and means for recognizing complex patterns,” U.S. Patent #3069654.

[4] Hirose, K. and Iwano, K. (1997). “A method of representing fundamental frequency contours of Japanese using statistical models moraic transition,” Proc. European Conf. Speech Communication and Technology, Rhodes, Greece, vol.1, pp.311–314.

[5] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2002). *The HTK Book, Version 3.2,* Cambridge University Engineering Department.

Chapter 10

REDUCTION OF DIFFUSE NOISE IN MOBILE AND VEHICULAR APPLICATIONS

Hamid Sheikhzadeh[1], Hamid Reza Abutalebi[2], Robert L. Brennan[1], and George H. Freeman[3]

[1]Dspfactory Ltd., 611 Kumpf Drive, Unit 200, Waterloo, Ontario, Canada N2V 1K8, [2]Electrical Engineering Dept., University of Yazd, Yazd, Iran; [3]Electrical and Computer Engineering Dept., University of Waterloo, 200 University Ave. West, Waterloo, Ontario, Canada N2L 3G1; Email: hsheikh@dspfactory.com

Abstract: In this chapter, we describe a hybrid subband adaptive speech enhancement system, implemented on an efficient ultra-low resource hardware platform utilizing oversampled generalized DFT filterbanks. Two analysis filterbanks decompose the two inputs (reference noise and noisy speech) into two sets of subband signals. In each subband, a subband adaptive filtering noise reduction block processes the two subband signals to reduce the noise producing a single signal which is followed by further noise reduction through Wiener filtering. Next, a synthesis filterbank converts the processed subband signals back into the time-domain. We have evaluated the performance of the hybrid noise reduction system in various real-life noise fields occurring in mobile and vehicular applications. Two closely spaced microphones make recordings in these noise fields. Signals from one microphone are used directly and represent the reference noise signal while signals from the other microphone are added to speech materials chosen from the TIMIT database before being used as the contaminated primary signal. It is demonstrated that all the noise recordings closely obey a diffuse noise field model. As the hybrid enhancement system is specifically designed to handle diffuse noise fields, it outperforms both the SAF and standard Wiener filtering in all sets of recordings. The superiority of the hybrid system is especially noted in the case of lowpass noise and intense noise conditions.

Keywords: LMS, Wiener filter, subband adaptive filter (SAF), oversampled filterbank, speech enhancement, diffuse noise, car noise, DFT filterbank, low-resource system.

1. INTRODUCTION

Adaptive noise cancellation through Subband Adaptive Filtering (SAF) has shown good performance when the two input noises are correlated [1]. However, many real-life noise fields in mobile and vehicular applications are correlated only at lower frequencies because they are approximately diffuse [2].

Diffuse noise fields are mathematically characterized through the spatial coherence function commonly used to specify the correlation of two noise signals x and y recorded at two input microphones in a noise field. The spatial coherence function is defined based on the cross- and auto-spectral densities as [3]:

$$\Gamma_{xy}^2 = \frac{|P_{xy}(f)|^2}{P_{xx}(f) \cdot P_{yy}(f)}. \qquad (1)$$

For 3-dimensional diffuse noise fields, the spatial coherence, obtained by averaging Eq. (1) over spherical coordinates, drops with frequency, following a $sinc^2(\cdot)$ of the form [2,3]:

$$\Gamma_{xy}^2 = \frac{\sin^2(2\pi fd/c)}{(2\pi fd/c)^2} = sinc^2(2fd/c), \qquad (2)$$

where c is the sound velocity ($c \cong 340$ m/s in air) and d is the distance between the two input microphones. Considering Eq. (2), it is obvious that SAF (and any adaptive noise cancellation method based on correlation cancellation) can only eliminate the noise in the lower frequency regions where there is a high correlation (coherence) between the two microphone signals.

To compensate for the inability of SAF to eliminate noise in diffuse noise fields, we have proposed a hybrid system integrating the SAF system and Wiener filtering (called SAFWF here), and have examined its performance in an isolated non-reverberant sound room [4]. Based on the promising results obtained, we further evaluate the performance of the SAFWF system in real-life mobile and vehicular noisy environments.

This chapter is organized as follows. The employed speech enhancement system is described in the next section. Section 3 describes the noise and speech materials used. System evaluations are reported in Section 4, and finally conclusions of this work are presented in Section 5.

2. EMPLOYED SPEECH ENHANCEMENT SYSTEM

The employed speech enhancement system is an oversampled Generalized DFT (GDFT) filterbank using a subband processing block consisting of a Subband Adaptive Filter (SAF) and a Wiener filtering sub-block to reduce noise in each subband. A Voice Activity Detector (VAD) is used to control both the adaptation in the SAF, and the noise spectrum estimation in the Wiener filter. The complete speech enhancement system (the SAFs, the Wiener filter, and the VAD) is efficiently implemented on an ultra-low resource oversampled WOLA filterbank detailed in [5]. We now describe various components of the enhancement system.

2.1 Subband Adaptive Filters for Noise Cancellation

Subband Adaptive Filters have become a viable choice for adaptive noise and echo cancellation. The SAF approach employs filterbanks to split time-domain inputs into a number of frequency bands, each serving as input to an adaptive filter. Subband signals possess reduced spectral dynamics and, due to their narrower bandwidth, may be decimated. Subband decomposition and decimation thus result in much "whiter" signals as input to a parallel bank of much shorter adaptive filters with better convergence behavior [6]. If critical sampling is used, aliasing distortion occurs that may be eliminated by employing either adaptive cross-filters between adjacent subbands or gap filters [6,7]. Systems with cross-filters generally converge slower and have higher computational cost, while systems employing gap filters produce significant signal distortion. OverSampled SAF (OS-SAF) systems, on the other hand, offer a simplified structure that, without employing cross-filters or gap filters, significantly reduce the aliasing level in subbands. In an attempt to minimize additional computational cost, a non-integer oversampling factor close to one $(1\langle OS\langle 2)$ is sometimes used. For many low-delay real-time applications including adaptive noise and echo cancellation, however, higher oversampling factors permit wide-range subband gain adjustments and the use of shorter windows with less stringent requirements [5]. Consequently, to avoid aliasing and other distortions, the solution of choice is to use an oversampling factor of two or more. However, oversampling degrades the convergence behavior of SAF systems (due to coloration of the subband signal) when the Normalized Least Mean Square (NLMS) algorithm is employed.

To improve the convergence rate and computation complexity of OS-SAF systems, we have proposed convergence improvement techniques [8] and have analyzed their effects on the system performance [9]. To further decrease the computation cost for low-resource implementations, partial update NLMS algorithms have been employed in combination with the convergence improvement techniques [10].

To improve the performance of SAF systems in diffuse noise fields, a hybrid system was proposed that takes advantage of the complementary characteristics of subband adaptive and Wiener filtering, resulting in a much higher noise reduction performance for diffuse noise fields [4]. Here we briefly introduce the employed OS-SAF system.

Shown in Figure 10-1 is the block diagram of the employed enhancement system. Two identical analysis filterbanks split the two inputs: the reference (noise) signal *x(n)* and the primary (noisy) signal *y(n)* into subband signals. After decimation by a factor of R , two subband signal sets $\{x_0(m), x_1(m), \cdots, x_{K-1}(m)\}$ and $\{y_0(m), y_1(m), \cdots, y_{K-1}(m)\}$ are obtained. Next, a subband processing block (denoted by SPB in Figure 10-1, described below) reduces the noise in each frequency subband. Finally, the synthesis filterbank combines the subband enhanced signals $\{z_0(m), z_1(m), \cdots, z_{K-1}(m)\}$ to obtain a time-domain output $z(n)$.

2.2 The DSP System

We employ highly oversampled GDFT uniform analysis/synthesis filterbanks based on Weighted OverLap-Add (WOLA). The WOLA filterbanks are optimally implemented on an ultra-low power hardware platform depicted in Figure 10-2.

The DSP portion consists of three major components: a WOLA filterbank coprocessor, a 16-bit DSP core, and an input-output processor (IOP). The DSP core, WOLA coprocessor, and IOP run in parallel and communicate through shared memory. The parallel operation of the system enables the implementation of complex signal processing algorithms in low-resource environments with low system clock rates. The system is especially efficient for stereo subband processing.

The core has access to two 4-kword data memory spaces, and another 12-kword memory space used for both program and data. The core provides 1 MIPS/MHz operation and has a maximum clock rate of 4 MHz at 1 volt. At 1.8 volts, 33 MHz operation is also possible. The system operates on 1 volt (i.e., from a single battery). The input-output processor is responsible for management of incoming and outgoing samples. It takes as input the speech signal sampled by the 16-bit A/D converter on the analog portion of the chip at a frequency of 8 kHz. The analog portion of the chip also applies a DC-cancellation filter to the speech signal. Through the DFT, the WOLA filterbank modulates a single prototype filter into *K* complex filters (*K / 2* unique bands due to Hermitian symmetry). Referring to Figure 10-1, each subband processing block is generally an adaptive filter working on a specific frequency band thus modeling a narrow frequency band of the combined acoustical and electrical transfer functions between the two microphones.

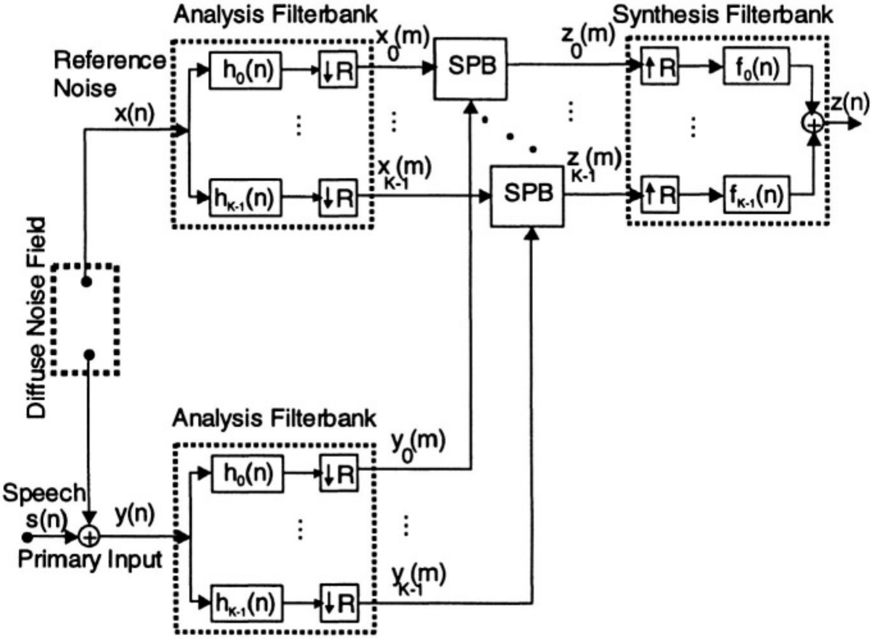


Figure 10-1. Block diagram of the SAF system with a Subband Processing Block (SPB) per subband.

2.3 Hybrid Adaptive-Wiener Filtering Enhancement Method

In the original SAF system, an adaptive filter is used as the subband processing block. In the SAFWF, this block is replaced with an adaptive filter cascaded with a Wiener filter as depicted in Figure 10-3. After elimination of the correlated noise components by the adaptive filter, the

Wiener filter further processes the error signal reducing the remaining uncorrelated noise. Since the quality of the error signal is already improved through adaptive filtering prior to Wiener filtering, it is expected that any introduced artifacts will be greatly reduced compared to Wiener filtering alone [4]. Objective evaluations reported in Section 4 confirm this expectation.

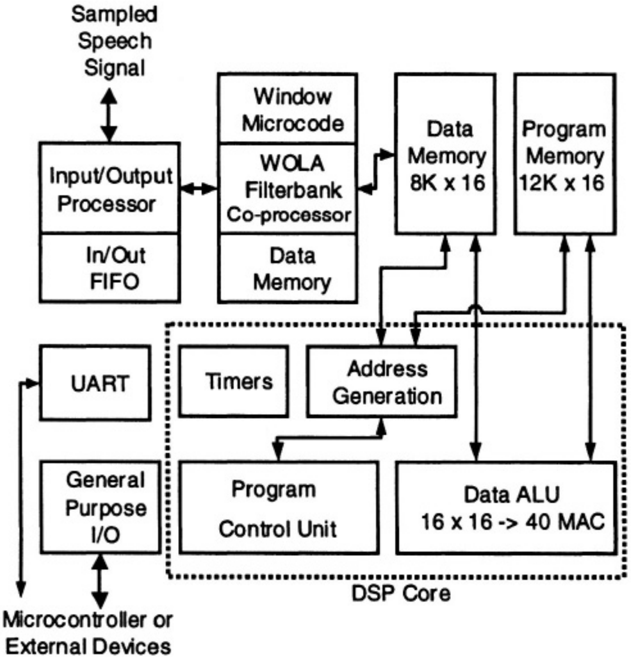


Figure 10-2. Block diagram of the DSP system.

Wiener filtering is implemented using a frequency-domain generalized Wiener filter [11]. As shown in Figure 10-3, the (subband) adaptive filter outputs are multiplied by a time-varying real gain $G_k(m)$; i.e., $Z_k(m) = G_k(m) \cdot E_k(m)$, where

$$G_k(m) = \left[1 - \frac{\beta \mid \hat{N}_k(m) \mid^{\alpha}}{\mid E_k(m) \mid^{\alpha}} \right]^{\frac{1}{\alpha}}, \tag{3}$$

and $\alpha = 1$, $\beta = 1.5$ are Wiener filter parameters that have been optimized for this application [4]. Also, $\hat{N}_k(m)$ is the uncorrelated noise spectrum estimated from $E_k(m)$ during speech pauses.

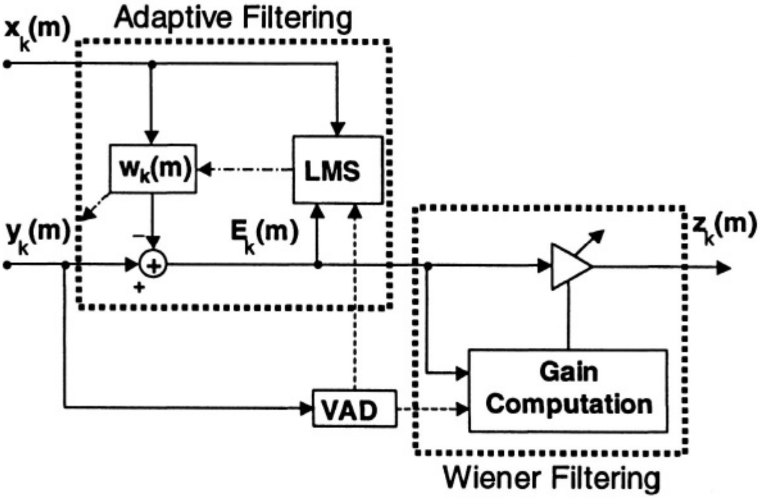


Figure 10-3. Block diagram of the subband processing block.

2.4 Voice Activity Detector

A Voice Activity Detector (VAD) has been employed to detect the noise-only portions of the primary (noisy) input. It is a modified version of the ETSI AMR-2 VAD [12] that has been implemented on the oversampled WOLA filterbank [13]. As depicted in Figure 10-4, the WOLA filterbank analysis results for the primary input are first grouped as a number (N_c) of channels and the energies of the channels ($E_i(m)$, $i = 1,2,\ldots,N_c$, m is the frame index) are estimated. Given an estimate of the background noise ($En_i(m)$, $i = 1,2,\ldots,N_c$), channel SNR ($\sigma_i(m)$, $i = 1,2,\ldots,N_c$) is estimated. A non-linear function maps the channel SNR to a voice metric *V(m)*. Channel SNR is also used to calculate a frame SNR and a long-term SNR. The voice metric and the long-term SNR provide primary parameters for the VAD decision. There is also a hangover mechanism in the VAD. A spectral deviation estimator measures the deviation between the frame subband energies and the long-term subband energies. When the deviation (averaged over subbands, $\Delta E(m)$), becomes very small, it may trigger a noise update under certain circumstances. An estimate of frame total energy ($E_{tot}(m)$) is also provided by the spectral deviation estimator. The noise energies are updated if the voice metric is less than a threshold. Otherwise, the spectral deviation is used to decide on a "forced noise update".

When a speech pause is detected, the subband adaptive filters are adapted, and the noise spectrum estimate of the Wiener filter ($\hat{N}_k(m)$ in Eq. (3) is updated.

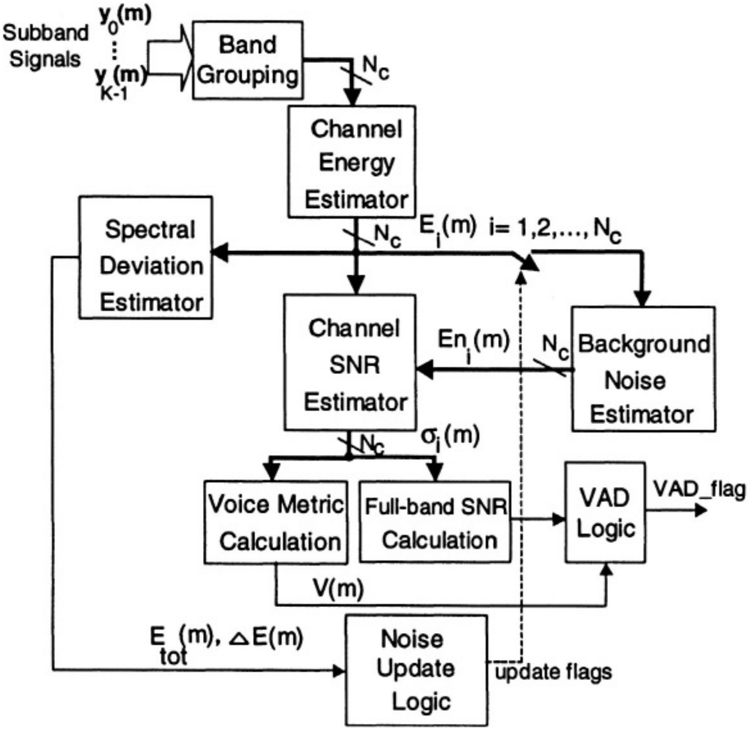


Figure 10-4. Block diagram of the employed VAD on the hardware platform.

3. NOISE & SPEECH FOR SYSTEM EVALUATION

The performance of the hybrid SAFWF system was evaluated employing several noises recorded by two input microphones in real-life situations. The sampling frequency was 16 kHz and the microphone spacing was set to

$d = 38$ mm (a typical value for boomless headsets). Recordings were done in the following situations:

- Sitting in a shopping mall (Sit-Mall1)
- Inside a working car parked next to a highway (HWY)
- Inside a moving car with open windows (CarWOpen)
- Inside a moving car with closed windows (CarWClose)

In each recording set, the first and second microphone inputs are considered as primary and reference noises, respectively. Figures 10-5(a)-(d) display the average Power Spectral Density (PSD) of the primary noise in each case. While all recorded noise source spectra are lowpass, the PSDs of the two Car noises (CarWOpen and CarWClose) fall much faster with frequency than the others. The effect of engine noise mostly appears as two local peaks at about 2.7 kHz and 5.4 kHz in Figures 10-5(b)-(d).

The spatial coherence curves of the first and second microphone signals (representing signals x and y in Eq. (1)) in the above recording situations are plotted in Figures 10-6(a)-(d), respectively. As depicted, the curves closely match the theoretical spatial coherence computed for diffuse noise by Eq. (2) with $d = 38$ mm. This observation is consistent with the diffuse noise assumption usually considered for most environmental noise fields in vehicular applications [2]. Also, due to its directivity, the engine noise acts more as a coherent source evident from the peaks in the coherence function at 2.7 kHz and 5.4 kHz (Figures 10-6(b)-(d)). This is especially evident for noises in a moving car with closed windows (Figure 10-6(d)).

Several sentences from the TIMIT [14] database were used as the speech material. For each set of the noise recordings, the primary noise was added to speech signal at 0 dB SNR and utilized as the noisy input.

4. HYBRID SPEECH ENHANCEMENT EVALUATION

Through subjective and objective evaluations, the performance of the SAFWF system is compared to that of the SAF approach. Also, we have examined the performance of single-microphone standard Wiener filtering (STDWF) method where the filter is directly applied to the primary noisy input.

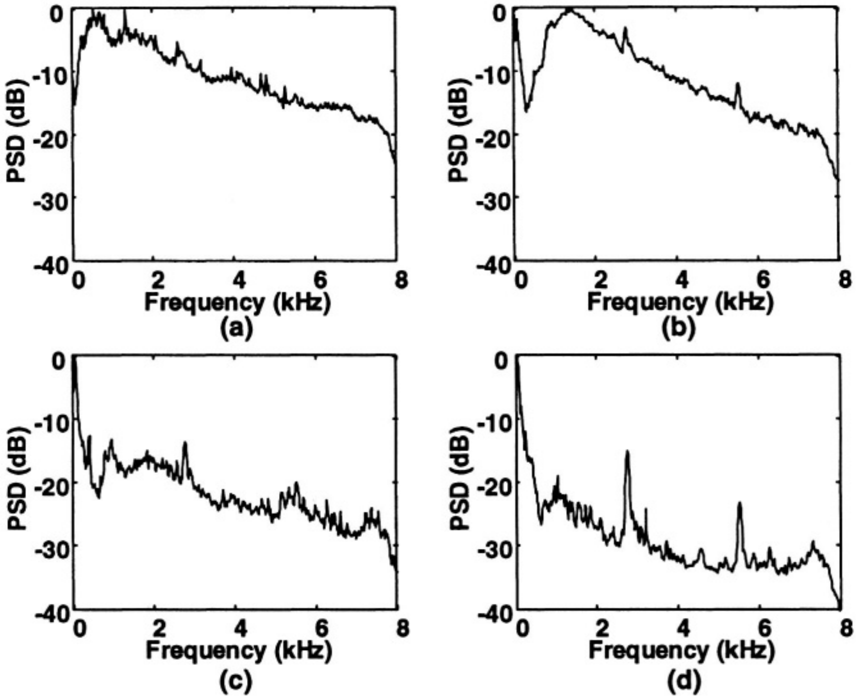


Figure 10-5. PSD of the noises recorded by the primary microphone in various situations: (a) Sit-Mall1, (b) HWY, (c) CarWOpen, (d) CarWClose.

To objectively measure the system performance, we use the Log Area Ratio (LAR) metric that has been shown to have the highest correlation with subjective assessments among all frequency-invariant distance measures [15]. Given the reflection (Partial Correlation, PARCOR) coefficients of order *L*, $\rho(m,l), l=1,\ldots,L$ of the m$^{\text{th}}$ frame, the corresponding Area Ratio (AR) is defined as [15]:

$$AR(m,l)=\frac{1+\rho(m,l)}{1-\rho(m,l)}\,. \tag{4}$$

Here, we have set the order of the reflection coefficients to $L = 16$. Given $AR_s(m,l)$ and $AR_z(m,l)$ for the m$^{\text{th}}$ frame of signals *s(n)* (clean speech) and *z(n)* (enhanced output) in Figure 10-1, the LAR distance between the m$^{\text{th}}$ frames of two signals is calculated as [15]:

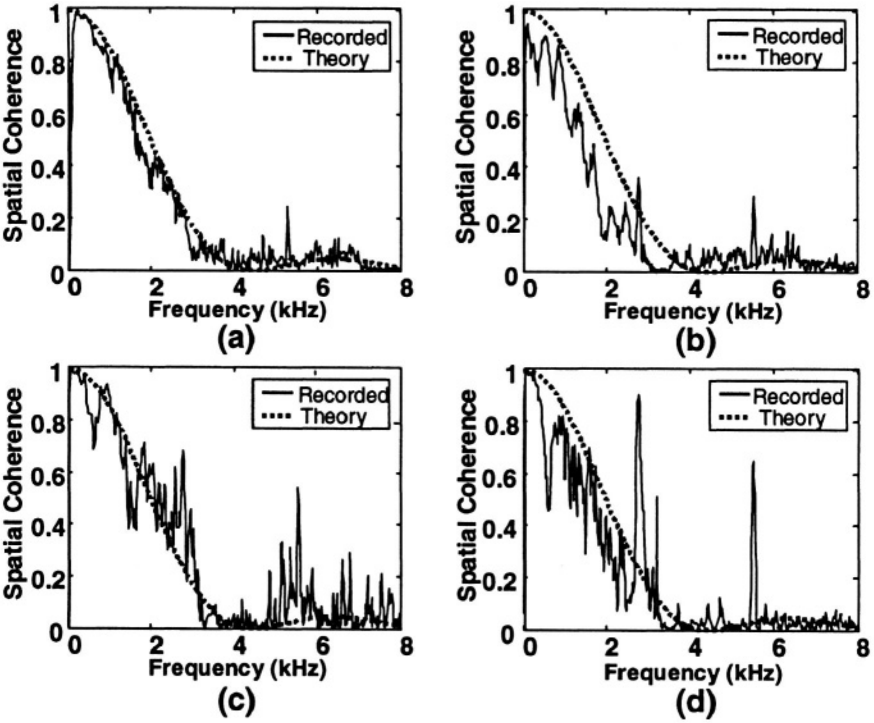


Figure 10-6. Spatial coherence of the noises recorded by two microphones in various situations: (a) Sit-Mall1, (b) HWY, (c) CarWOpen, (d) CarWClose.

$$LAR_{sz}(m) = \left\{ \frac{1}{L} \sum_{l=1}^{L} \left| 20\log_{10}\left[\frac{AR_s(m,l)}{AR_z(m,l)}\right] \right|^2 \right\}^{\frac{1}{2}}. \tag{5}$$

In order to remove frames with unrealistically high LAR distances, we compute the overall LAR distance by first discarding frames with the top 5% LAR values, and then averaging Eq. (5) over the remaining frames (as suggested in [16]).

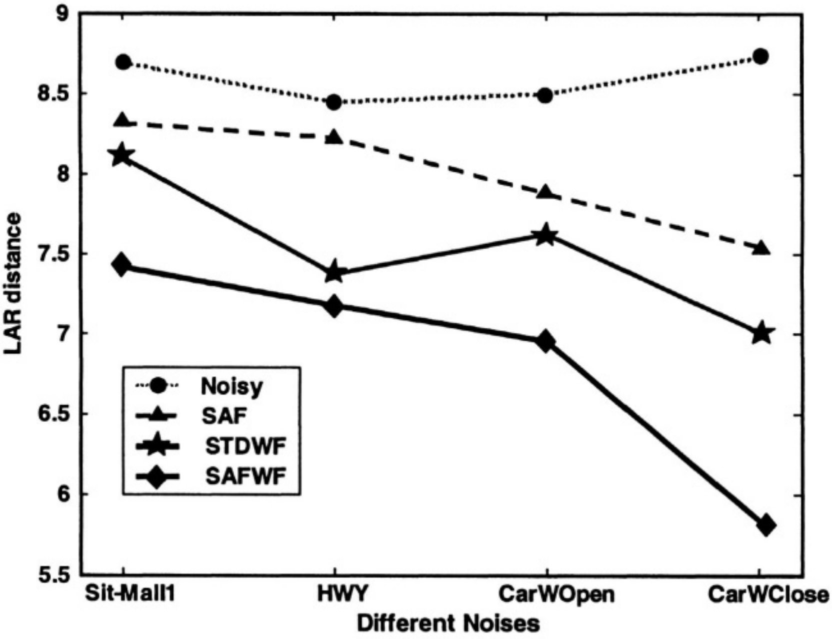


Figure 10-7. LAR-distances between the clean input and various output signals: unprocessed output (Noisy), outputs of the SAF, STDWF and SAFWF techniques in four different noisy environments.

The results of objective evaluations are shown in Figure 10-7. For reference, the LAR distance between the clean input (signal *s*(*n*) in Figure 10-1) and the unprocessed noisy output (signal *z*(*n*) in Figure 10-1 with all SPBs inhibited) is also presented. The low LAR-distance improvement obtained by the SAF indicates that this method has difficulty rejecting noise in diffuse noise environments. Evidently, SAF has better performance in the third and forth environmental conditions (CarWOpen and CarWClose). This can be justified by considering Figures 10-5 and 10-6 as follows: 1) In the CarWOpen and CarWClose cases, the noise spectrum is dominated by lowpass and coherent components. This results in improved noise reduction of the important lowpass components by adaptive filtering. 2) In particular, there are highly coherent engine-related noises in these two cases that are efficiently cancelled by the subband adaptive filters.

As is evident from Figure 10-7, the hybrid system (SAFWF) outperforms both the SAF and the STDWF systems for all four test sets. Especially in CarWOpen and CarWClose cases, the SAFWF method produces better LAR distance improvements. This demonstrates that an improvement in the adaptive filter performance leads to better performance of the Wiener filter

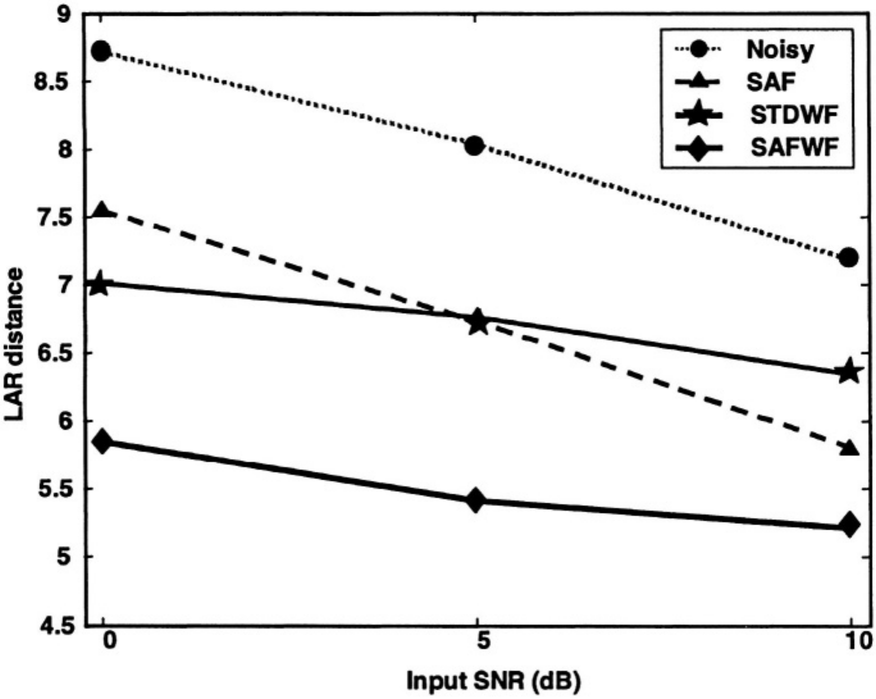


Figure 10-8. LAR-distances distances between the clean input and various output signals: unprocessed output (Noisy), outputs of the SAF, STDWF and SAFWF techniques for CarWClose case and different input SNRs.

since Wiener filters typically generate less speech distortion at high input SNRs.

In order to examine the effect of input SNR, we have repeated the objective assessments for the fourth set of noise recordings (CarWClose) at input SNRs of 0, 5, and 10 dB. As depicted in Figure 10-8, the SAFWF offers better performance in all cases. Considering LAR improvements of the SAFWF (relative to the LARs for the unprocessed output) at various SNRs (3, 2.5 and 2 for SNRs of 0, 5, and 10 dB, respectively) the superiority of the SAFWF is more evident at low input SNRs. Also, notice that the SAF performance improvement is almost the same for different SNRs. This is expected as the adaptive noise canceller by design, removes the correlated noise components.

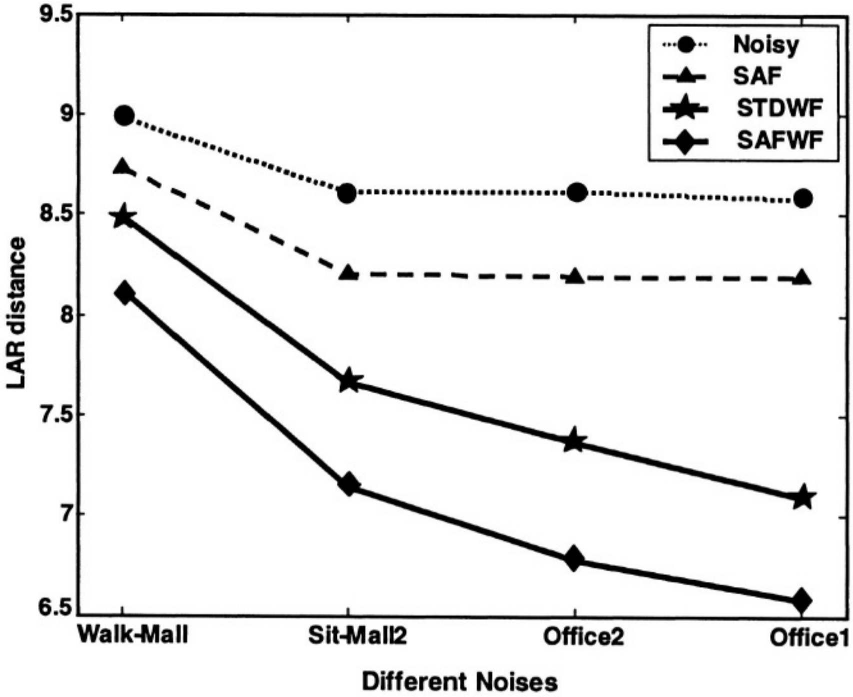


Figure 10-9. LAR-distances distances between the clean input and various output signals: unprocessed output (Noisy), outputs of the SAF, STDWF and SAFWF techniques in four new recording environments.

To further verify the system performance, we repeated the objective LAR tests using four other noise sets at 0 dB SNR. Two were recorded while walking (Walk-Mall), and sitting (Sit-Mall2) in a shopping mall. Two different office noise sets (Office1 and Office2) were also recorded. The recording methods were exactly the same as those used in Section 3. The objective evaluation results (keeping the same system parameters as in previous evaluations of Figure 10-7) are depicted in Figure 10-9 where the noises (from left to right) are sorted according to the increasing severity of the lowpass nature. As evident from Figure 10-9, the results are consistent with those presented in Figure 10-7.

Also, we have done some informal listening tests confirming the objective assessments. The artifacts produced by the STDWF technique are considerably reduced by applying the Wiener filter after the adaptive filtering.

5. CONCLUSION

Experimental results confirm the diffuse model for most noise fields in vehicular and mobile applications. In this research, we evaluated the performance of a hybrid subband adaptive and Wiener filtering structure for noise reduction in diffuse fields.

The hybrid system benefits from the advantages of adaptive filters in coherent bands and also utilizes the Wiener filter to remove uncorrelated components. By first improving the SNR as best as possible through correlated noise reduction using subband adaptive filtering, the inherent artifacts of standard Wiener filtering are considerably reduced.

Objective and subjective assessments confirm the superiority of the hybrid system for noise reduction in different real-life vehicular and mobile fields. Considering the spectral characteristics of the employed noises, it is clear that there are larger improvements for lowpass noise sources particularly at low SNRs.

REFERENCES

[1] K. Tam, H. Sheikhzadeh, and T. Schneider "Highly oversampled subband adaptive filters for noise cancellation on a low-resource DSP system," *Proc. ICSLP-2002*: *Inter. Conf. Spoken Lang. Processing,* pp. 1793-1796, Denver, CO, Sept. 2002.

[2] M. M. Goulding, "Speech enhancement for mobile telephony," *IEEE Trans. Vehic. Technol.,* vol. 39, pp. 316-326, Nov. 1990.

[3] M. Tohyama, H. Suzuki, and Y. Ando, *The Nature and technology of acoustic space,* Orlando, FL: Academic, 1995.

[4] H. R. Abutalebi, H. Sheikhzadeh, R. L. Brennan, and G. H. Freeman, "A hybrid subband system for speech enhancement in diffuse noise fields," *IEEE Signal Processing Letters,* vol. 11, No. 1, pp. 44-47, Jan. 2004.

[5] R. Brennan, and T. Schneider, "A flexible filterbank structure for extensive signal manipulation in digital hearing aids," *Proc. IEEE Int. Symp. Circuits and Systems,* vol. 6, pp. 569-572, Jun. 1998.

[6] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: analysis, experiments and application to acoustic echo cancellation," *IEEE Tran. Signal Processing,* vol. SP-40, pp. 1862-1875, Aug. 1992.

[7] J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine,* pp. 14-37, Jan. 1992.

[8] H. R. Abutalebi, H. Sheikhzadeh, R. L. Brennan, and G. H. Freeman, "Convergence improvement for oversampled subband adaptive noise and echo cancellation," *Proc. Eurospeech,* Geneva, Switzerland, Sept. 2003.

[9] H. Sheikhzadeh, H. R. Abutalebi, R. L. Brennan, and J. Sollazzo, "Performance limitations of a new subband adaptive system for noise and echo reduction," *Proc. 10th IEEE Inter. Conf. on Electronics, Circuits and Systems (ICECS 2003),* Sharjah, UAE, Dec. 2003.

[10] H. Sheikhzadeh , H. R. Abutalebi, R. L. Brennan, K. R. L. Whyte , and E. Chau, , "Sequential LMS for low-resource subband adaptive filtering: oversampled implementation and polyphase analysis," *Proc. XII European Sig. Proc. Conf., EUSIPCO 2004,* Vienna, Austria, Sept. 2004.

[11] J. S. Lim, and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proc. IEEE,* vol. 67, pp. 1586-1604, Dec. 1979.

[12] ETSI, "Universal Mobile Telecommunication Systems (UMTS); Mandatory Speech Codec speech processing functions, AMR speech codec; Voice Activity Detector (VAD) (3GPP TS 26.094 version 4.0.0 Release 4)". Eur. Telecommun. Standards Inst., Sophia-Antipolis, France, ETSI TS 126 094 V4.00 (2001-03).

[13] E. Cornu, et al. "ETSI AMR-2 VAD: evaluation and ultra low-resource implementation," *Proc. ICASSP,* vol. 2, pp. II - 585-588, Apr. 2003.

[14] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," NIST, 1993.

[15] S. R. Quackenbush, T. P. Barnwell III, and M. A. Clements, *Objective measures of speech quality,* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[16] J.H.L. Hansen, and B.L. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," *Proc. ICSLP,* vol. 7, pp. 2819-2822, Dec. 1998.

Chapter 11

SPEECH ENHANCEMENT BASED ON F-NORM CONSTRAINED TRUNCATED SVD ALGORITHM

Guo Chen, Soo Ngee Koh and Ing Yann Soon

School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore 639798 Email: esnkoh@ntu.edu.sg

Abstract: Traditional singular value decomposition (SVD) based speech enhancement algorithms are usually limited by the use of a fixed order of retained singular values which may not be optimal for time-varying noise corrupted speech signals. In this chapter, we propose the use of a Frobenius-norm (F-norm) constrained truncated (FCTSVD) algorithm in an analysis-by-synthesis procedure for choosing the appropriate order of retained singular values for speech enhancement. It allows for self-adaptation in time and for different noise and noisy speech characteristics. Also, it leads to the best approximation of original speech in terms of SNR. The proposed algorithm has been tested and compared with a traditional SVD algorithm for different noise types and levels. Simulation results show that it achieves higher SNR improvements for both additive white noise and colored noise as compared to a traditional SVD algorithm.

Keywords: Speech enhancement, singular value decomposition (SVD), Frobenius-norm.

1. INTRODUCTION

The use of speech processing systems for voice communication and speech recognition is becoming more and more common. This is largely due to the availability of low cost digital signal processors and memory chips. Among all the speech processing research efforts, the problem of enhancing speech degraded by additive broad-band noise is still an active research topic. During the past three decades, many single channel speech enhancement algorithms have been proposed. More recent published algorithms include variants of spectral subtraction [1] and amplitude estimation methods [2], methods based on all-pole modeling [3], enhancement using discrete cosine transformation (DCT)[4] or two-dimensional Fourier transform [5], schemes based on constructive-destructive additive noise[6], and signal subspace methods [7]. In this paper, we propose a new algorithm, called F-norm constrained truncated SVD (FCTSVD) algorithm, to solve the problem of how to automatically choose the appropriate order of retained singular values in an SVD scheme.

2. THE SVD AND SIGNAL SUBSPACE ESTIMATION

Let $Y = [y(0), y(1), \cdots y(L-1)]^T$ be an observed noisy signal vector with L samples and we assume that the noise is additive and uncorrelated with the signal, i.e. $Y = X + W$, where $X = [x(0), x(1), \cdots x(L-1)]$ and W represent the original and noise signal, respectively. We can construct the following MxN M Hankel matrix H_Y from Y as done in [7], where $L = M + N - 1$ *and* $M > N$. Correspondingly, H_Y can also be written as

$$H_Y = H_X + H_W \tag{1}$$

where H_X and H_W represent the Hankel matrices derived from X and W, respectively.

According to the SVD theory, there exist orthogonal matrices $U_Y \in \Re^{MxN}$ and $V_Y \in \Re^{MxN}$ such that $H_Y = U_Y \Sigma_Y V_Y^T - \sum_{i=1}^{N} u_i \sigma_i v_i^T$, where $\Sigma_Y = diag(\sigma_1, \sigma_2, \cdots, \sigma_N) \in \Re^{NxN}$, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0.$ The nonnegative diagonal elements of Σ_Y are called singular values of H_Y. Usually, it is convenient to partition the SVD of H_Y using the first P singular values of H_Y as follows:

$$H_Y = \begin{bmatrix} U_{Y1} & U_{Y2} \end{bmatrix} \begin{bmatrix} \Sigma_{Y1} & 0 \\ 0 & \Sigma_{Y2} \end{bmatrix} \begin{bmatrix} V_{Y1}^T \\ V_{Y2}^T \end{bmatrix} \tag{2}$$

where $U_{Y1} \in \Re^{MxP}$, $\Sigma_{Y1} \in \Re^{MxP}$, and $V_{Y1} \in \Re^{NxP}$. We make three assumptions here: (1) The signal is orthogonal to the noise in the sense that $H_X^T H_W = 0$; (2) The noise is white: $H_X^T H_W = \sigma_W^2 I$; (3) The smallest singular value of Σ_{Y1} is larger than the largest singular value of Σ_{Y2}, i.e., $\sigma_P > \sigma_{P+1}$. Generally, the minimum variance (MV) estimate is used in noise reduction [7].

Given the matrices H_Y, H_X and H_W as in (1), There exists a matrix $G \in \Re^{NxN}$ which minimizes $\min_{G \in \Re^{NxN}} \|H_Y G - H_X\|_F^2$, where $G = (H_Y^T H_X)^{-1} H_Y^T H_X$. Under these three assumptions, the MV estimated of H_X can be derived as,

$$\hat{H}_X = U_{Y1} \Sigma_{Y1}^{-1} (\Sigma_{Y1}^2 - \sigma_W^2 I) V_{Y1}^T \tag{3}$$

The last expression can also be denoted as

$$\hat{H}_X = U_{Y1} (F_{MV} \Sigma_{Y1}) V_{Y1}^T, \tag{4}$$

using the following PxP matrix filter,

$$F_{MV} = Diag\left((1 - \frac{\sigma_W^2}{\sigma_{Y,1}^2}), (1 - \frac{\sigma_W^2}{\sigma_{Y,2}^2}), \cdots, (1 - \frac{\sigma_W^2}{\sigma_{Y,P}^2}) \right). \tag{5}$$

Since $\hat{H}_X$ of H_X does not have the Hankel structure, it is necessary to make a Hankel matrix approximation to $\hat{H}_X$. A simple procedure, as described in [7], for restoring the Hankel structure is to average along the anti-diagonals of $\hat{H}_X$ and put each average value as a common element in the corresponding diagonal of a new Hankel structured matrix with the same dimension. In the meantime, it is noted that the choice of P takes an important role in the whole speech enhancement scheme in that a small value of P results in information loss, while a big value leads to a matrix not absolutely noise free.

3. F-NORM CONSTRAINED CRITERION

In order to obtain the appropriate order for the best reconstruction, an F-norm constrained function used in an analysis-by-synthesis procedure is proposed. Let $\hat{H}_X^{(s)}$ represent the estimated matrix of H_X from the first s singular values of H_Y, i.e.

$$\hat{H}_X^{(s)} = U_{Y1}^{(s)}(F_{MV}\Sigma_{Y1}^{(s)})(V_{Y1}^{(s)})^T \tag{6}$$

where F_{MV} is the *sxs* matrix filter as described in (5). We define an F-norm constrained function (FCF), denoted by Φ, as follows:

$$\Phi^{(s)} = \|H_Y\|_F^2 - \left\|\hat{H}_X^{(s)}\right\|_F^2 - \|H_W\|_F^2, \quad \text{for } s = 1,2,\ldots,N\,. \tag{7}$$

It can be proved that the FCF is a monotonically decreasing function with increasing value of the variable s. According to the F-norm definition and the assumptions, we have

$$\Phi^{(s)} = \|H_X\|_F^2 - \left\|\hat{H}_X^{(s)}\right\|_F^2 \tag{8}$$

Obviously, we also have,

$$\left\|\hat{H}_X^{(s+1)}\right\|_F^2 \geq \left\|\hat{H}_X^{(s)}\right\|_F^2, \tag{9}$$

due to the use of a new singular value in the reconstruction process of the estimated Hankel matrix with additional power. From (8) and (9), it is clear that the FCF is a monotonically decreasing function.

Equation (8) states that the F-norm of the reconstructed $\hat{H}_X$, with $s < P$, is less than that of H_X coming from the clean signal. This is because

$\hat{H}_X$, does not contain all the harmonic information of the clean signal x. On the other hand, when $s > P$, the F-norm of the reconstructed $\hat{H}_X$ is larger than that of H_X due to noise. This implies that $\Phi^{(s)}$ crosses zero at $s = P$. However, the non-correlation assumption is not a true condition in practice, especially in the case of short data records. Thus a small bias may exist in the values of $\Phi^{(s)}$. This slight deviation can be solved by investigating the change of the difference values of the FCF as follows. Based on the definition of $\Phi^{(s)}$, we have

$$\Phi^{(s+1)} - \Phi^{(s)} = \left(\frac{\sigma_{Y,s}^2 - \sigma_W^2}{\sigma_{Y,s}} \right)^2 - \left(\frac{\sigma_{Y,s+1}^2 - \sigma_W^2}{\sigma_{Y,s+1}} \right)^2 \tag{10}$$

Note that $\sigma_{Y,s}^2$ consists of only the noise component when $s \geq P+1$. This implies that the difference value of the FCF converges to 0, i.e. $\Delta\Phi^{(s)} = | \Phi^{(s+1)} - \Phi^{(s)} | \to 0$. In true condition, the value of $| \Phi^{(s+1)} - \Phi^{(s)} |$ converges to a small value. Thus, we can use a threshold value ξ_0. If the value of $| \Phi^{(s+1)} - \Phi^{(s)} |$ is initially less than ξ_0 with the increase of s, we can then obtain the wanted order of retained singular values as follows:

$$P = \arg \underset{1 \leq s \leq N}{first} \left\{ \Phi^{(s+1)} - \Phi^{(s)} | \leq \xi_0 \right\} - 1 \tag{11}$$

Note that the selected P by F-norm constrained criterion leads to the best approximation of original speech in terms of signal-to-noise ratio (SNR). Let $\hat{X}$ be a frame of the reconstructed signal and it consists of the original signal X and error signal ε, i.e. $\hat{X} = X + \varepsilon$. Assuming the means of $X, \hat{X}$ and ε are zero, and that the error signal ε is statistically independent of the clean signal X, the SNR of the reconstructed signal is then given as follows:

$$SNR = 10 \log_{10} \left(\frac{E(X^2)}{\left| E(\hat{X}^2) - E(X^2) \right|} \right) \tag{12}$$

where E is the statistical expectation operator. Next, we will prove that the SNR becomes maximum when the FCF reaches the point $s = P$. Let

$M_X = (1/(NxM))\|H_X\|_F^2$, according to the F-norm definition, we have $M_X = \frac{1}{N}[\frac{1}{M}\sum_{i_0=0}^{M-1} x^2(i_0) + \frac{1}{M}\sum_{i_1=0}^{M-1} x^2(i_1) + \cdots + \frac{1}{M}\sum_{i_{N-1}=0}^{M-1} x^2(i_{N-1})]$. As X is a wide-sense stationary stochastic process in the short-time period [0, L –1], the autocorrelation of X, R_X, does not depend on the placement of the time origin, i.e., $R_X(t, t-\tau) = E[x(t)x(t+\tau)] = R_X(\tau)$. Hence, it is clear that $R_X(0) = \underset{i_0=0\to M-1}{E}[x(i_0)x(i_0)] = \underset{i_1=1\to M}{E}[x(i_1)x(i_1)] = \cdots = \underset{i_{N-1}=N-1\to M+N-2}{E}[\ x(i_{N-1})x(i_{N-1})\ ]$, and that $M_X = R_X(0) = E(X^2)$. Similarly, we also have $M_{\hat{X}} = E(X^2)$. Correspondingly, it can be shown that

$$SNR = 10\log_{10}\left(\frac{\|H_X\|_F^2}{\left|\|H_X\|_F^2 - \|\hat{H}_X^{(s)}\|_F^2\right|}\right) \tag{13}$$

Meanwhile, we also know from the above discussion that $|\|H_X\|_F^2 - \|\hat{H}_F^{(s)}\|_F^2|$ attains its minimum at $s = P$, and consequently the SNR of the reconstructed signal $\hat{X}$ attains its maximum.

4. THE FCTSVD ALGORITHM

Based on the above discussion, we can formulate the FCTSVD algorithm for the case of white noise in the following steps.

(1). Estimate noise vectors W from silence periods in the observed speech signal.

(2). Form the Hankel matrix H_Y from the observed signal.

(3). Compute the SVD of H_Y.

(4). Initialize the order of retained singular values of $\hat{H}_X$, i.e., $s = 0$.

(5). Let $s = s + 1$ and reconstruct the estimated matrix of $H_X, \hat{H}_X^{(s)}$, using the first s singular values of H_Y.

(6). Compute the FCF. If $s = 1$ then return to ***Step 5***.

(7). Compute the difference values of the FCF, i.e. $\xi^{(s)} = \Phi^{(s)} - \Phi^{(s-1)}$.

(8). Decide the appropriate order. If $|\xi^{(s)}| \le \xi_0$, then $P= s-1$ else return to ***Step 5.*** Based on our experiments, we have used the threshold value of $\xi_0 = 0.0098$.

(9). Compute the enhanced speech signal $\hat{X}$ from $\hat{X}_X^{(P)}$.

If the additive noise W is colored, a pre-whitening transformation is applied to the data matrix. Assuming the sample Hankel matrix H_W of the noise signal is known, then the corresponding Cholesky factorization or QR decomposition is given by $H_W^T H_W = R^T R$ or $H_W = QR$, where $R \in \Re^{NxN}$ is the upper triangular Cholesky factor and $Q \in \Re^{NxN}$ has orthonormal columns $Q^T Q = I_N$. The implementation algorithm is exactly the same as that described in the above procedure except for two extra steps at the beginning and the end of the procedure. They are described below:

(3). Compute the QR decomposition of H_W and perform a pre-whitening of H_Y, (i.e. $H_W = QR$, and $H_Z = H_Y R^{-1}$.)

The following Steps 4-9 are exactly the same as the above described Steps 3-8 except that H_Y, is replaced by H_Z.

(10). Perform a de-whitening of $\hat{H}_X^{(P)}$, (i.e. $\tilde{H}_X^{(P)} = \hat{H}_X^{(P)} R$.)

Finally, the reconstructed speech signal, $\hat{X}$, is computed from $\tilde{H}_X^{(P)}$ by arithmetic averaging along its anti-diagonals.

5. SIMULATION RESULTS

5.1 Order Determination

The performance of the proposed F-norm constrained algorithm is initially examined using the reconstruction of several voiced and unvoiced speech signals. The consecutive reconstructions of the signals have been produced with increasing value of s while the corresponding values of the FCF, as well as the SNRs of the reconstructed signals, are calculated. The frame length and analysis order of SVD are chosen, based on our simulations, to be 200 and 21, respectively, i.e., $M = 180$, $N = 21$.

The results of reconstructing a frame of voiced and unvoiced speech signals embedded in white noise with SNR=0dB are given in Fig. 11-1 and Fig. 11-2, respectively. As shown in Figs. 11-1 and 11-2, the dots of the FCF form the convergent curves and the value of $\Delta|FCF|$ converges to a very small value (approximate zero). In our simulation study, the convergent threshold ξ_0 is set to be 0.0098. Correspondingly, the order in the reconstruction of voiced speech signal is chosen to be 9 with around 7.25dB maximum SNR, while the order of unvoiced speech signal is chosen to be 13 with around 5.01dB which is close to the maximum SNR. It is clear that the maximum of the SNR curve occurs at the selected order determined by the FCF although there is a slight bias against unvoiced speech signal due to its noise like characteristics.

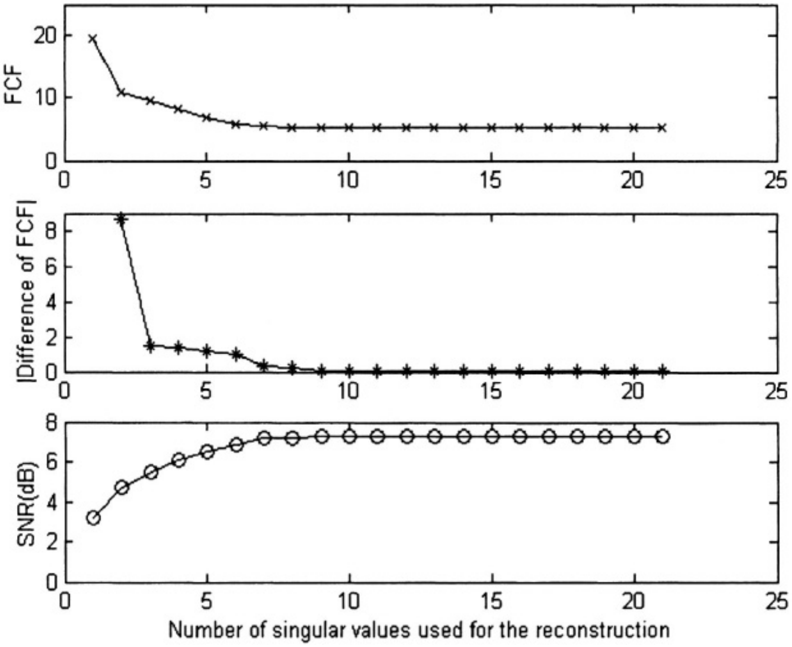


Figure 11-1. Reconstruction of the Voiced Speech Signal.

5.2 Evaluation of Enhancement Performance

The performance evaluation of the proposed algorithm has been tested and compared with the truncated SVD (TSVD) algorithm for MV estimation reported in [7]. The fixed order of the TSVD algorithm is selected to be 14 by which the best enhanced performance is attained as described in [7]. Four speech utterances of two male and two female speakers from the TIMIT database are used in our study. Four different background noises are taken from the Noise-92 database. Segmental SNR (SegSNR) improvement is used for assessing the enhancement performance. The results are shown in Table 11-1. It can be seen that the proposed algorithm leads to higher SegSNR improvements than TSVD for all noise types and levels.

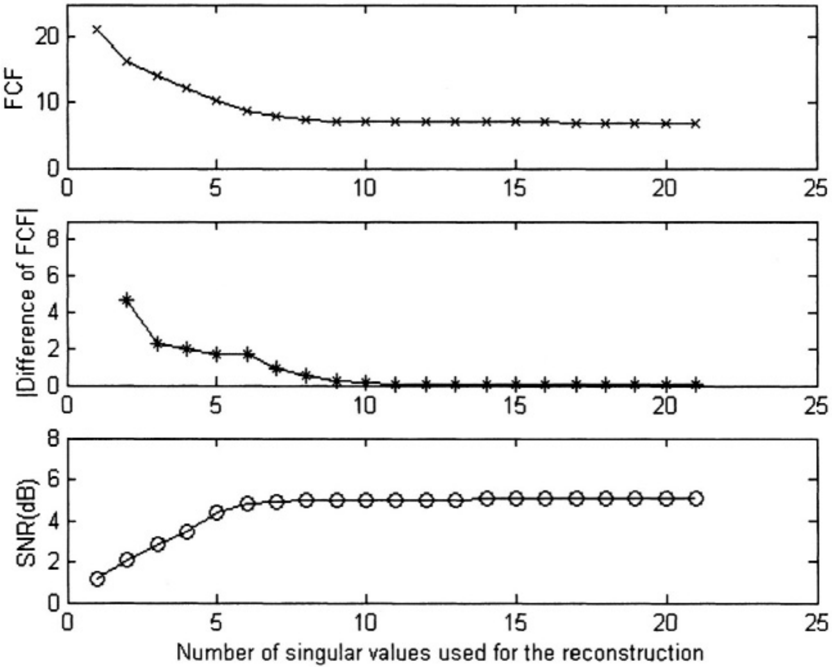


Figure 11-2. Reconstruction of Unvoiced Speech Signal.

Algorithms	White Gaussian	Pink	F16 cockpit	Babble
TSVD	6.59	6.23	6.14	4.86
FCTSVD	8.09	8.00	7.79	6.99

Table 11-1. The average SegSNR Improvements for different noise types (dB)

CONCLUSIONS

In this chapter, the use of a Frobenius-norm (F-norm) constrained truncated (FCTSVD) algorithm in an analysis-by-synthesis procedure has been investigated for choosing the appropriate order of retained singular values for speech enhancement. It allows for self-adaptation in time and for different noise and noisy speech characteristics. Also, it leads to the best approximation of original speech in terms of SNR. The proposed algorithm has been tested and compared with a traditional SVD algorithm for different noise types and levels. Simulation results show that it achieves higher SNR improvements for both additive white noise and colored noise as compared to a traditional SVD algorithm.

REFERENCES

[1] P. Scalart and J. V. Filho, “Speech enhancement based on a priori signal to noise estimation,” Proceedings of IEEE ICASSP-96: Inter. Conf. Acoustics, Speech and Signal Processing, pp. 629-632, Atlanta, GA, 1996.

[2] I. Y. Soon and S. N. Koh, “Improved noise suppression filter using self-adaptive estimator of probability of speech absence,” Signal Processing, vol.75, pp. 151-159, 1999.

[3] J. H. L. Hansen, and M. A. Clements, “Constrained iterative speech enhancement with application to speech recognition,” IEEE Transactions on Signal Processing, vol.39, pp. 795-805, 1991.

[4] I. Y. Soon, and S. N. Koh, “Noisy speech enhancement using discrete cosine transform,” Speech Communication, vol.24, pp. 249-257, 1998.

[5] I. Y. Soon, and S. N. Koh, “Speech enhancement using two dimensional Fourier transform,” to appear in IEEE Transactions on Speech and Audio Processing, 2003.

[6] I. Y. Soon, and S. N. Koh, “Low distortion speech enhancement,” IEE Proceedings Vision, Image and Signal Processing, vol.147, no.3, pp. 247-253, 2000.

[7] S. H. Jensen, P. C. Hansen, S. D. Hansen and J. A. Sorensen, “Reduction of Broad-Band Noise in Speech by Truncated QSVD,” IEEE Trans. on Speech and Audio Processing, vol.3, no.6, pp.439-448, 1995.

Chapter 12

VERBKEY - A SINGLE-CHIP SPEECH CONTROL FOR THE AUTOMOBILE ENVIRONMENT

Rico Petrick[1], Diane Hirschfeld[1], Thomas Richter[1], Rüdiger Hoffmann[2]
[1] *voice INTER connect GmbH,Dresden, Germany.* [2]*Laboratory of Acoustics and Speech Communication, Dresden University of Technology, Germany*
Email: petrick@voiceinterconnect.de

Abstract: The article deals with a novel speech recognizer technology which has the potential to overcome some problems of in-car speech control. The *verbKEY* recognizer bases on the Associative-Dynamic (ASD) algorithm which differs from established techniques as HMM or DTW. The speech recognition technology is designed to run on a 16 bit, fixed point DSP platform. It enables high recognition performance and robustness. At the same time, it is highly cost efficient due to its low memory consumption and its less calculation complexity. Typical applications such as dialling, word spotting or menu structures for the device control are processed by the continuous, real-time recognition engine with an accuracy higher 98% for a 20 words vocabulary. The article describes a hardware prototype for command & control applications and the measures taken to improve the robustness against environmental noises. Finally, the authors discuss some ergonomic aspects to obtain a higher level of traffic safety.

Keywords: Automatic speech recognition, Associative-Dynamic classifier (ASD), robustness, telephone application, discriminative optimization

1. IN-CAR SPEECH CONTROL - REQUIREMENTS AND ISSUES

Controlling in-car devices and hands-free communication by speech commands is a major safety issue in modern cars. Speech control definitely decreases the information load of the driver so that he can dedicate his attention to the traffic and the driving instead of searching knobs and looking around in the cockpit at high speed.

The state of the art, however, looks somewhat different. There are very few speech control solutions that can meet the high requirements of an acoustically difficult car environment.

Unsatisfactory functionality is still the major reason for the comparably poor customer's acceptance of speech control. One could improve the acceptance with high level speech recognizers and sophisticated user interfaces. On the other hand, speech control in cars is expected to provide a high accuracy, to be robust against noise and environmental variations, but also to be very cost efficient. Typical requirements for speech control systems in particular apply to cars:

- High recognition accuracy,
- Use of command words or command phrases,
- Fixed set of speaker-independent commands, and
- Programmable set of speaker-dependent commands.

Furthermore, some special requirements for in-car applications can be defined:

- Speed-independent background noise characteristics,
- Low-cost embedded solution,
- 'Push to talk' to prevent false alarms,
- Known (mostly stationary) speaker-microphone distance.

In the following section, a novel approach to command word recognition is presented and set into relation to established techniques such as HMM or DTW. The following section shows, how a hands-free solution is implemented based on this core algorithm. Finally, measures are presented to improve the module's robustness against typical background noises. The chapter is finished with a discussion on future developments.

2. RECOGNIZER TECHNIQUES

2.1 Basic approaches for speech recognition

Hidden Markov Models (HMM) are the state of the art technology and widely used in practical applications. In isolated word recognition, a HMM recognizer offers a recognition rate near to the human performance. It is suitable for difficult recognition scenarios (e.g. fluent speech with a large vocabulary or spontaneous speech). Usually, HMMs demand a high modeling effort and a floating-point arithmetic for the necessary computational precision. The component costs to perform HMMs are high and often oversized for simple control applications with a small number of commands.

A further speech recognition technique is based on artificial neural networks (ANN). An ANN is suitable to handle static patterns and self-adapting processes. Low-cost solutions sometimes employ ANN techniques. Except using the more complex TDNN approach (Time Delay Neural Network), these solutions usually do not achieve satisfactory recognition accuracy.

Recognizers using the principle of Dynamic Time Warping (DTW) require less computational precision and modeling effort than HMM. A major drawback is the increasing memory demand, if DTW recognizers are speaker-independently trained. Generally, DTW recognizers can achieve a similar recognition accuracy than HMM recognizers.

2.2 ASD algorithm

The patented Associative-Dynamic (ASD) recognizer was developed at the Dresden University of Technology to provide a very cost-efficient and simple recognizer alternative [1]. It requires ultra-low resources and it is suitable to most command and control tasks in mobile applications. It can be implemented at low-cost processor platforms. Several measures support the memory reduction and the low processing load:

- *Reduced feature dimensions by a discriminative network without loss in classification accuracy.* An associative network at the front-end of the classifier transforms the primary feature vectors x (describing the object to classify and coming from the analyzer in equidistant time intervals), into secondary feature vectors y with reduced dimension and improved discrimination properties. By this transformation, the input pattern is adapted to the statistical characteristics of the reference knowledge of the classifier. The transformation weights are optimized for a given recognition task in a training step by an evolutionary procedure [1].
- *Task-dependent distance operators.* There is a choice of distance operators by which optimal performance of the classifier for a given recognition task and under varying accuracy conditions (fixed- vs. floating point) can be achieved. Local distances are calculated by applying the distance operator on each input- and reference vector pair.

The following, dynamic network aggregates the time-varying local distances *d* to a temporally varying distance-vector *g*. In case of continuous recognition, hypotheses are selected from the search space according to their scores and stored to a time-varying n-best list.

- *Efficient pruning on commands, words and on the acoustic level.* For continuous word recognition, intermediate hypotheses are stored in n-best lists ordered by their scores. If the command syntax allows it, hypothesis-trees are grown during the recognition process, and new hypotheses are only started if a possible word end is found. So the search space of word and subword units can be reduced substantially and the search is only conducted through a subset of reference models. On the acoustic level, score based pruning reduces the number of active grid points in the matching process [2]. All these measures reduce processing load and allow the implementation of the recognition engine even on simple, low performant processor platforms.

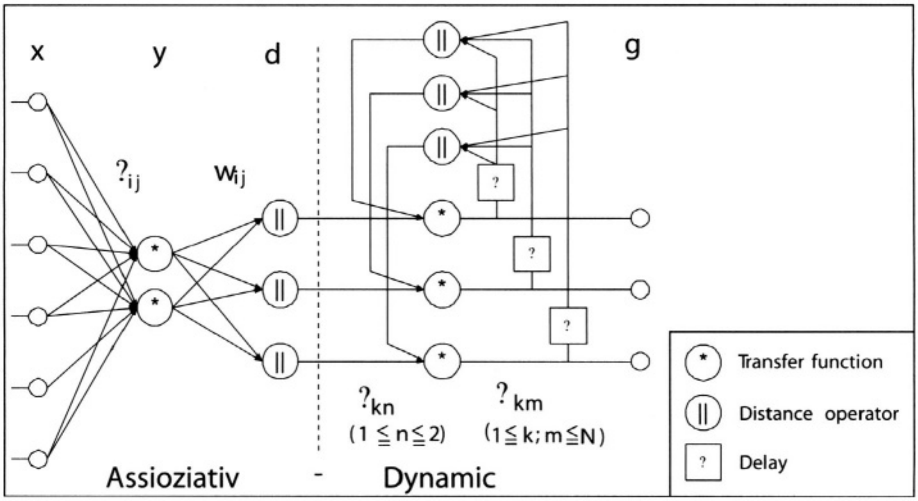


Figure 12-1. Associative Dynamic (ASD) classifier in network representation, x primary feature vector, y secondary feature vector, d local distance, g aggregated, optimal distance during matching process.

- *Temporal compression of reference patterns.* Temporal redundancy is avoided by compressing the reference patterns for the basic acoustic units in a way, that the remaining reference states represent only stable and - in terms of classification - relevant parts of the original pattern. For reasons of discriminative power, the temporal structure of the original pattern is preserved, too.

- *Discriminative training and lexicon optimization.* The simplicity of the recognizer structure allows fast retraining of the recognition engine for a different classification task. A set of efficient, automatic tools supports the design of the reference knowledge under the focus of high recognition accuracy and robustness against environmental noise.

For isolated word or command phrase recognition, an evolutionary tool is optimizing the associative feature extraction part of the ASD classifier. The fitness criteria, expressing the quality of a recognizer individually, are high recognition accuracy as well as high discriminative ability between different classes. Evolutionary algorithms outperform conventional mathematical feature extraction methods [5,6] in that they are able to overcome local optima and find the global optimum.

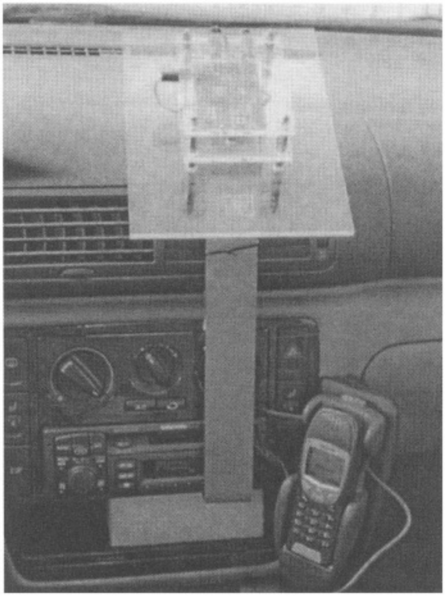


Figure 12-2. Hands-free demonstrator on the base of verbKEYv1.6

A drawback of this method is its high resource consumption, because the optimum often is found in several optimization attempts (5-10) after some 100 generations of 50 recognizer individuals each were evaluated. At the moment, this costly process is implemented in a distributed evaluation network, where one computer runs the evolutional engine, which sends the parameters to optimize over the LAN. All other computers connected to the network run 1 to 3 recognizer individuals (clients), and send the results of the evaluation back to the evolution. The training and test data is accessed by each client over the network. In this way, a speaker independent word recognition rate of 98.6 % can be reached in the absence of noise (see Section 4.3).

3. TARGET APPLICATION AND DIALOGUE

The target application is a telephone control with combined speaker independent commands for hands-free telephone operation, and speaker dependent commands for name dialling. The application is fully operable by voice, because the user is guided through all menus by high quality voice prompts.

To increase the robustness of the application the recognizer is embedded in an ergonomic dialog including voice prompts and ‘push to talk’. There are 30 speaker independent commands and 30 speaker dependent commands (plus corresponding actions like a telephone number for each command) available in the lexicon. Commands are ordered in submenus to enable functions like user dependent training (storage of new names into the lexicon), dictation of phone numbers (collection of number chains, repetition and navigation in the number chain).

The high sound quality of the voice prompts ensure a high acceptance of the speech control by the user. Prompts are stored in a scalable memory and are generated by an application-specific word-unit synthesizer. So even large and well tuned dialogs can be stored with minimum memory requirements. The dialogue is designed very flat to minimize the distraction of the driver from traffic.

4. ROBUSTNESS

Recognition accuracy loss mainly occurs at the speaker microphone path. Some important influences on the quality of the received speech signal and methods to overcome them are listed below:

- *Microphone type*. Electret capacitor microphones with directional characteristics should be used in cars to attenuate side noises and keep the system costs low. Microphone arrays can be used for beam forming. At present, the higher costs caused by hardware and processing power are generally accepted because of the benefits to noise robustness. In this chapter, only a single channel approach was used.

- Speaker-microphone distance. The microphone is mounted at the A-Pillar or at the interior mirror. There the Speaker-microphone distance can be supposed in a range of 20 to 50 cm for the driver. In those distances the level of the speech signal should be sufficient if the user speaks loud enough to obtain at least a Signal to Noise Ratio (SNR) of 5, better 10 dB.
- Surrounding noise. Surrounding noises can be eliminated by the built in automatic noise reduction of the speech recognizer. The noise reduction adapts automatically to the present background noise. Stationary or slowly varying noises such as fan, engine or road noises can be handled. No adaptation is possible for sudden or strong transient noises such as screen wiper, radio sound or conversational noise.
- Conversational noise *(babel speech).* Background conversational noise is difficult to separate from the commands of the driver, because the conversation has the same characteristics as the commands. Microphones with directional characteristics or beam forming microphone arrays possibly attenuate the disturbing speech signal.

4.1 Automatic noise reduction

The Automatic Noise Reduction (ANR) employs the principle of spectral subtraction and is included in the signal chain of the analyzer after the Fourier transform (FFT) of the input signal.

Adaptation of the ANR. For spectral subtraction it is necessary to know the spectral noise characteristics. Therefore a voice activation detector in the time domain (TVAD) marks the pause intervals where no speech is present. The pause decision in the TVAD is based on evaluation of the signal energy in relation to two adaptive decision thresholds for speech and silence [3]. Spectra in pauses are averaged to estimate the noise spectrum (Figure 12-3). This is done using a low pass filter [4].

$$|N_{est}(f)| = \rho \cdot |N_{est}(f, l-1)| + (1-\rho) \cdot |X_{in}(f)| \tag{1}$$

where $|N_{est}(f)|$ is the estimated noise spectrum, $|X_{in}(f)|$ is the short time microphone spectrum in pauses, ρ is the adaptation factor, and l the spectra index. Fast, transient changes such as a door clap do not affect the estimation but slow changes such as the motor sound during car acceleration do.

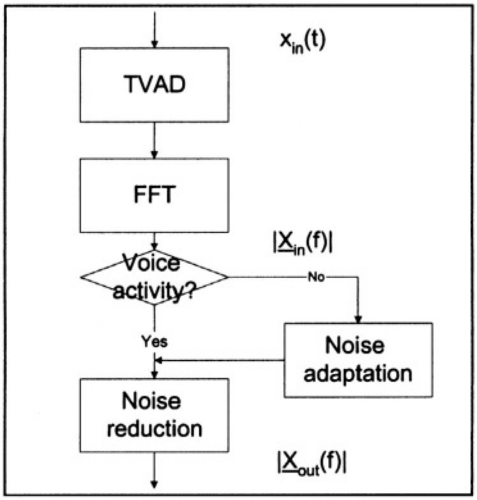


Figure 12-3. Parts of the preprocessing with noise reduction.

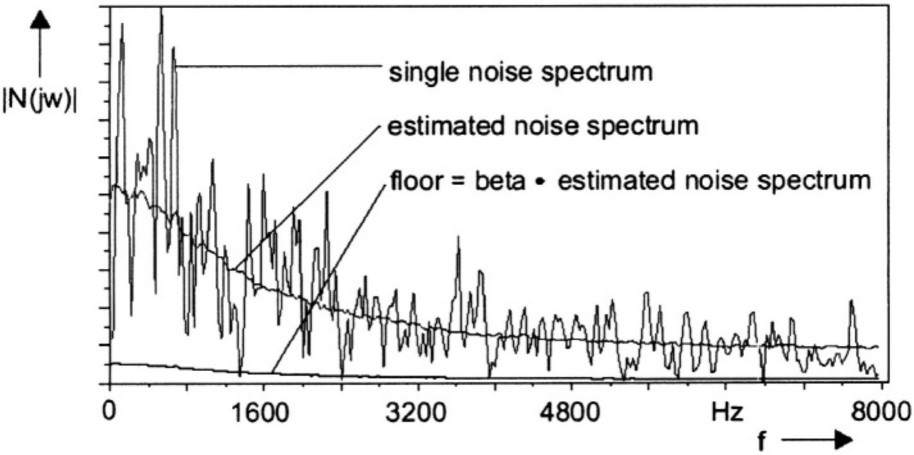


Figure 12-4. Pause spectrum, estimated noise spectrum and flooring level.

Spectral subtraction: The estimated noise spectrum is subtracted from the incoming (disturbed) signal corresponding to the following two equations:

$$|X_s(f)| = |X_{in}(f)| - \alpha \cdot |N_{est}(f)| \tag{2}$$

$$|X_{out}(f)| = \begin{cases} |X_s(f)|; \; |X_s(f)| > \beta \cdot |N_{est}(f)| \\ \beta \cdot |N_{est}(f)|\,; |X_s(f)|; \le \beta \cdot |N_{est}(f)| \end{cases} \tag{3}$$

where $|X_s(f)|$ is the mathematical result of traditional spectral subtraction, $|X_{out}(f)|$ is the result after flooring, α determines the intensity of the spectral subtraction, and β the flooring factor. Suitable values for α are 1.5 ... 2.2. It is also possible to have an adaptive behavior of $\alpha(\mathrm{SNR})$ for the entire frequency band or in subbands α (SNR, f) [4].

After subtraction, flooring is applied. Flooring is needed to avoid musical tones but also to have a sensible level of comfort noise. Experiments to find the optimal value for the flooring factor resulted in $\beta = 0.1$. β can also be determined in an adaptive way as a function of SNR or by a different setting for subbands.

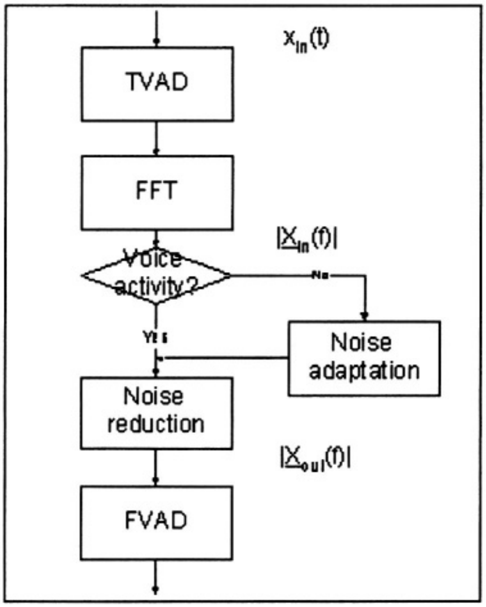


Figure 12-5. . Combination of VAD approaches in the time and frequency domain lead to improved word detection accuracy

4.2 VAD under different noise conditions

A prerequisite for a good noise suppression is a robust voice activity detection (VAD), that adapts to varying background noise conditions. The energy threshold based VAD approach in the time domain (TVAD), as shown in Fig. 12-3. has very robust adaptation characteristics. On the other hand, the approach has shown limited performance on weak, fricative sounds at word beginnings in the presence of stationary, loud background noise, that is very similar to the energy distribution of the fricatives.

Therefore, a VAD was implemented in the frequency domain (FVAD), that combined several features (delta energy, smoothed sum of energy and peak to average ratio) from the ETSI-approach [7] to a voice activity decision. This frequency domain approach performed very well under clean conditions, but it showed to be sensitive against background noise.

In order to gain from the advantages, a combined approach of a time-domain and a frequency-domain VAD was tested as shown in Fig. 12-5. The TVAD was used to provide pause boundaries for a robust noise adaptation. After the noise reduction, the SNR was increased, and the cleaned speech signal provided stable input conditions for the FVAD. The FVAD is then used to refine and confirm the pause decisions of the TVAD.

The performance of the VAD approaches was evaluated on a database of 20 command phrases of 13 subjects with varying background noise. There were three noise conditions: silence, medium and high level fan noise (27, 14 and 5 dB SNR). The performance of the VADs was measured as difference between manually and automatically determined word boundaries. The results for the noise conditions for TVAD only, FVAD only and combined approach are given in Figure 12-6.

As Fig. 12-6 shows, the absolute deviation between the automatically and manually derived word boundaries (in ms) are divided in the categories that are given below the figures. Also in Fig. 12-6, the category "No Labs" means that no boundary was found automatically. The first 3 categories display an acceptable performance of the VAD.

4.3 Recognition accuracy in the presence of noise

A total of 12 experiments with 37 different speakers and a test vocabulary of 17 command phrases have been accomplished. Tables 12-1, 12-2, and 12-3 show the word recognition rates (WRR) for 3 different environmental conditions and 4 speaker microphone distances (SMD). For each experiment 111 realizations per command phrase have been tested (17*111=1887 commands per experiment). The SNR dependency of the recognition accuracy can be well studied.

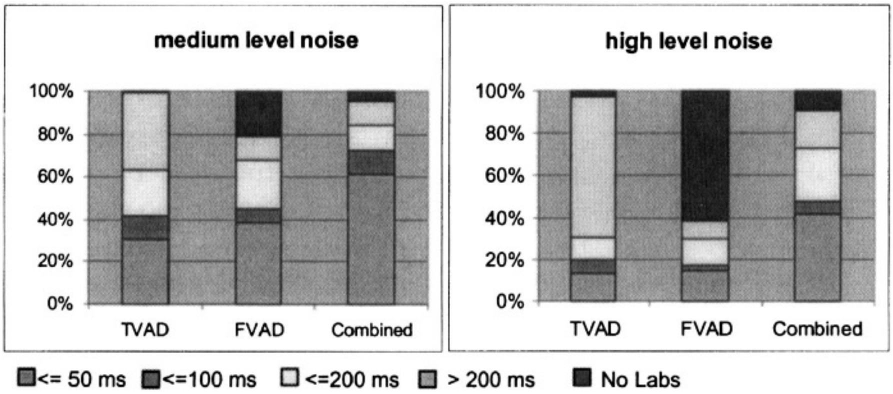


Figure 12-6. Performance of TVAD, FVAD and combined approach under different noise conditions. Deviations from manually labeled word boundaries. Medium noise level: 14 dB SNR, high noise level: 5 dB SNR

SMD	0.4 m	1 m	1.5 m	5 m
SNR	32 dB	26 dB	24 dB	17 dB
WRR	98.6 %	97.6 %	97.2 %	94.2 %

Table 12-1. Quiet room background noise

SMD	0.4 m	1 m	1.5 m	5 m
SNR	14 dB	8.3 dB	7 dB	0 dB
WRR	98.0 %	95.4 %	93.9 %	85.1 %

Table 12-2. Fan noise (level 2)

SMD	0.4 m	1 m	1.5 m	5 m
SNR	10 dB	2.3 dB	1.9 dB	-4 dB
WRR	96.6 %	92.7 %	91.6 %	81.5 %

Table 12-3. Fan noise (maximum level)

The SNR values are derived by the following equation:

$$SNR = 10 \cdot \log_{10} \frac{(X - N)}{N} \tag{4}$$

where X is the mean power of disturbed signal (presence of speech - averaged over entire utterance), N is the power of the noise (no presence of speech), and SNR is the resulting subtraction coefficient.

Experiments with speech material from a running car at different speed conditions (50 km/h, 90 km/h, 120 km/h) for streets with a plain surface showed similar recognition results like the fan-sound experiments.

For streets with cobbled surface, the WRR decreased by about 5 % per condition, because the background noise was not that stationary anymore. Therefore, the detection accuracy of voice activity boundaries was decreased, and this led - besides a poorer noise estimation - to a worse word recognition rate.

A second improvement was gained by using a stationarity detector in the VAD and by switching the word detectors' parameters corresponding to the estimated stationarity of the background noise in order to make the detection for voice events more or less sensitive.

5. ISSUES OF HARDWARE DEPENDENT IMPLEMENTATION

For the porting and optimal functionality on a low cost hardware platform, several requirements have to be met. The optimizations aim at a reduction of memory, processing load and need for computational accuracy. Intending to run the ASD recognizer on several platforms, it was implemented in ANSI C, which makes the code highly portable. First it was implemented on a 32 bit floating point DSP with 60 MHz, later 16 bit fixed point DSP were used. Processor specific changes of the code base where applied rarely in order to speed core routines (assembly subroutines) or to interface hardware components such as codec, displays, UART or CAN. Vendors of car equipment can use the speech recognizer as a single OEM-board (see Figure 12-7) or as a sub-application on their own processor.

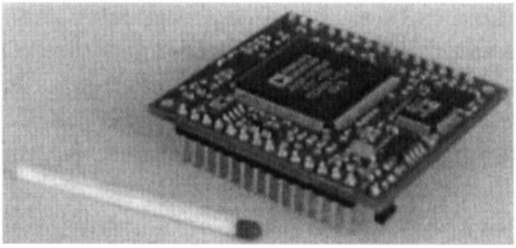


Figure 12-7. Hardware OEM-module vicCORE2195G3

The algorithms were implemented with a strict code data separation. Since the software reference is an experimental system and contains by far more optional algorithms than used in a given application, a technology was developed for the easy porting of all the control structures and knowledge bases of the recognizer. A number of automatically generated C-headers, which contain the whole initialized bunch of control structures and knowledge bases, are included automatically in the compilation of the DSP code. In this way, a fast and efficient porting is achieved, without loss in accuracy and without the need of type conversions from one platform to another.

6. SUMMARY

A new speech recognition technology was presented in this chapter. It was shown, how this technology was applied in a typical car application and which steps were taken to provide the necessary robustness under adverse acoustic conditions.

It should be lined out, that technology is not the only key component for the success of a product. Since modern technical products have implemented more and more complex functions, information sometimes overload users. Reading of extensive manuals is needed, and this often leads to the effect that the user only knows the features that are necessary for a basic device operation.

The design of the suitable user interface more and more decides on the success of a product. Speech interfaces allow a much more powerful and efficient user interface than available in conventional devices. The replacement of buttons by speech commands is only a small part of the possible advantages of speech interfaces.

In the future, collaboration between speech interface designers, device vendors and customers should be much closer to explore fully the new ergonomic opportunities of speech driven user interfaces. Efficient dialogs lead the operator automatically to the desired function. The goal is a self-explaining composition of devices with a managing intelligence, which interacts with the internal states of the devices as well as with the user via speech and buttons. A well-designed interface in the car minimizes distraction from traffic during device operation and makes the study of user manuals unnecessary.

REFERENCES

[1] T. Rudolph, *Evolutionary Optimization of Fast Command Recognizers,* (in German), Phd thesis, Dresden University of Technology, 1998.

[2] A. Noll A. Paesler H. Ney, D. Mergel, "Data-driven search organisation for continuous speech recognition," *IEEE Trans. Signal Processing,* vol. 40, pp. 272–281, 1992.

[3] U. Koloska T. Richter R. Petrick D. Hirschfeld, J. Bechstein, "Development steps of a hardware recognizer with minimal footprint", (in German), *Proc. 13th Conf. on Electronic Speech Signal Processing (ESSV), Dresden,* pp. 182–189, 2002.

[4] W. Hess P. Vary, U. Heute, *Digital Speech Signal Processing,* (in German), Teubner, Stuttgart, 1998.

[5] G. Ruske: *Automatische Spracherkennung – Methoden der Klassifikation und Merkmalsextraktion,* München: Oldenbourg Verlag, 1988.

[6] Fukunaga: *Introduction to Statistical Pattern Recognition,* San Diego: Academic Press, 1990.

[7] ETSI EN 301, V7.1.1: *Voice activity detector (VAD) for AdaptiveMulti-Rate (AMR) speech traffic channels*, General description (GSM 06.94 version 7.1.1 Release 1998)

Chapter 13

REAL-TIME TRANSMISSION OF H.264 VIDEO OVER 802.11B-BASED WIRELESS AD HOC NETWORKS*

E. Masala,[1] C. F. Chiasserini,[2] M. Meo,[2] J. C. De Martin[3]
[1]*Dipartimento di Automatica e Informatica;* [2]*Dipartimento di Elettronica;* [3]*IEIIT-CNR Politecnico di Torino, Italy Email: demartin@polito.it*

Abstract This chapter aims at evaluating a number of Quality of Service (QoS) indices of a real-time video transmission over an 802.11b ad hoc wireless network. Video is coded according to the state-of-the-art ITU-T H.264 encoder and its transmission is simulated by means of the *ns-2* network simulator. Objective quality measurements are presented. Moreover, the impact of different parameters — both at the encoder and at the MAC level —, of background interfering traffic and of the number of relay nodes, is studied, showing the various trade-offs involved.

Keywords: Real-time video, IEEE 802.11 wireless local area networks, ad hoc wireless networks.

1. INTRODUCTION

The great success of the IEEE 802.11b technology for wireless local area networks (WLANs) [1] is creating new opportunities for the deployment of advanced multimedia services. Important applications such as telephony, videoconferencing and audiovisual streaming are on path to move to Wireless Local Area Networks (WLANs), creating a complex, yet highly attractive scenario, where users will be able, at least in principle, to seamlessly switch from typ-

*This work was supported in part by CERCOM, the Center for Multimedia Radio Communications, Torino, Italy, http://www.cercom.polito.it.

ically expensive, wide-area coverage to cheaper, higher-bandwidth local and micro-local networks.

Two different 802.11b WLAN scenarios are possible: *with infrastructure* or *ad hoc*. The former includes an access point, i.e., a central controller that is typically connected to the wired network, and several wireless stations that can communicate with the access point only. The latter consists of a peer-to-peer network where wireless stations can directly communicate with each other, thereby allowing a low-cost communication system that supports mobile users and a dynamic network environment.

In this work, we focus on ad hoc networks supporting real-time multimedia applications. A couple of examples come to mind: users in an airport hall equipped with laptops and forming a network, who wish to download from a server news or entertainment programs; or a video-surveillance sensor network, where each sensor controls a portion of the area of interest and sends the information to a far-away node producing the output video of the whole area.

Real-time multimedia transmission over ad hoc WLANs poses several challenges. Radio bandwidth is limited, and propagation conditions over the radio channel may significantly vary in time, often leading to quite large error rates. In addition, ad hoc networks usually exploit multihop communications that enable wireless stations to reach a distant destination by a sequence of short-range communication links. On the one hand, this allows nodes to overcome their limited radio range and avoid the large battery consumption involved in long-range transmissions. On the other hand, multihop communications involve an additional delay in traffic delivery, which increases with the number of hops between source and destination. Besides, in the case of 802.11b-based networks, sources must contend for the radio channel whenever they wish to transmit. Thus, they experience access delays that may significantly degrade perceptual quality.

Solutions to these problems have recently been the focus of several works [14, 2, 10, 4, 11, 9]. In [14], the authors study the performance of the 802.11b DCF and PCF Medium Access Control (MAC) schemes for an integrated H.263 and data traffic scenario, in a WLAN with infrastructure. The work in [2] describes the design of an architecture for H.263+ multicast video on ad hoc 802.11b WLANs and presents some experimental results. In [10] source coding is combined with Forward Error Correction (FEC) coding and an automatic repeat request (ARQ) technique to efficiently support unicast and multicast real-time video streaming in an 802.11b WLAN with infrastructure. The studies in [4, 9, 11] specifically address error resilience of real-time multimedia streams in ad hoc networks, although their main focus is on traffic routing. In [4] the authors describe the design and the demonstration of a set of simple routing protocol mechanisms, and the performance they obtained. In [9] and [11] the mesh structure of an ad hoc network is exploited to allow multiple paths between a source and a destination, thus improving reliability of video transmissions. To further enhance error recovery, an ARQ technique is applied. Also, in [9] the authors explore the possibility to employ layered coding as well as multiple description coding. However, none of these works considers an 802.11b ad hoc scenario supporting both real-time video and data traffic, and investigates the effects of interfering traffic.

The objective of our work is to study the transfer of video sequences over wireless ad hoc networks using the 802.11b technology, and globally optimize the parameters involved in a real-time video transmission, ranging from video encoding and packetization to the MAC interface parameters. Moreover, we evaluate the possibility to provide good quality real-time video under multi-hop network scenarios.

We consider the state-of-the-art ITU-T H.264 [8] video encoder and configure it to optimally match the ad hoc network scenario, as well as to adapt to varying channel conditions. Standard video test sequences are packetized according to the H.264 Network Adaptation Layer (NAL) for transmission using the RTP/UDP/IP protocol stack. Error resilience tools provided by the H.264 standard are also configured and adapted to the characteristics of the 802.11b wireless medium. We consider the presence of interfering data traffic carried by TCP connections. The quality perceived by the video user at the receiver is objectively evaluated, using the PSNR as a distortion measure. By means of the *ns* [15] network simulator, we simulate several network conditions, which include various different channel conditions and different numbers of hops in the path between source and destination.

2. IEEE 802.11b AD HOC NETWORKS

We consider an ad hoc network composed of stationary wireless stations, using the IEEE 802.11b technology.

The 802.11b standard operates in the 2.4 GHz frequency bands, enabling transmission rates ranging between 1 and 11 Mbps. IEEE 802.11 cards transmit at a constant power, achieving a transmission range of up to hundreds of meters.

Two wireless stations are said to be within range and said to be neighbors of each other if they can receive the each other's transmission. Every station can employ multihop transmissions to transfer information toward its final destination; also, it always behaves in a cooperative fashion accepting to act as a router and relay traffic destined to other stations. For instance, if station s needs to

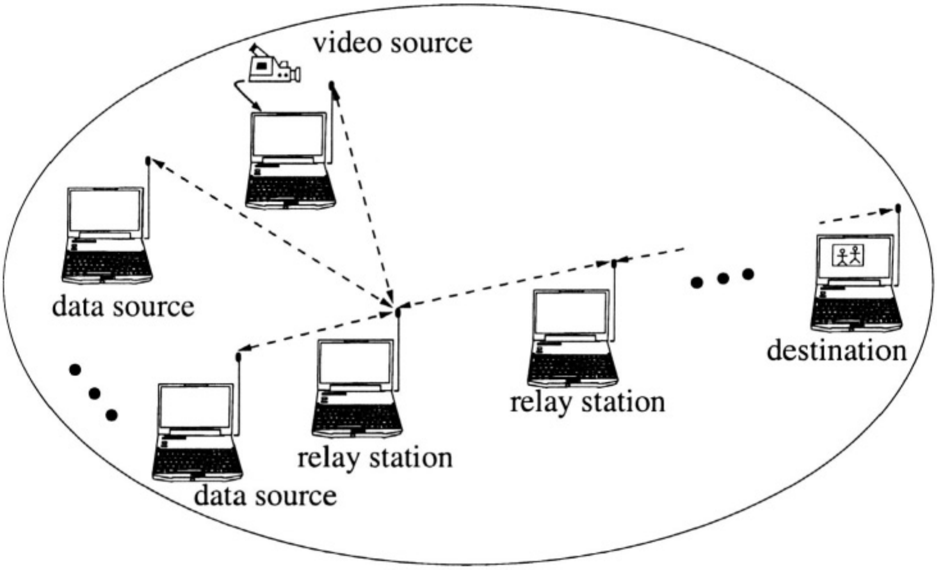


Figure 13-1. The 802.11b ad hoc network scenario.

send traffic to d and d is not within the range of s, then the information is sent to one of s's neighbors, say r. Node r will forward that information to its neighbor, and so on, until it reaches the destination, d. In this example, r acts as a router.

As a reference scenario, we consider an ad hoc network including one station generating video traffic, and up to eight stations which generate data traffic. The overall network scenario is shown in Figure 13-1. All sources are associated with the same destination and use the same relay stations to deliver their traffic to the destination. While all sources are in the radio proximity of each other, only the first relay station can communicate with the destination node, possibly by using other intermediate relay nodes. A routing algorithm specifically designed for wireless ad hoc networks, such as DSR [6] or AODV [12], can be used to establish a route for each source-destination pair.

At the MAC and physical layers, all stations employ the 802.11b functions. In particular, we assume that all stations can transmit at 11 Mbps and access the channel by using the basic 802.11b MAC scheme, the so-called Distributed Coordination Function (DCF) [1]. According to DCF, wireless stations wishing to transmit a MAC Protocol Data Unit (MPDU) employ a CSMA/CA mechanism, based on the listening-before-transmitting criterion. A station's transmission may fail either because of the bad propagation conditions over the channel or because two or more stations transmit at the same time and collide. However, an ARQ scheme is implemented: in the case of failure, a transmission is repeated until a maximum number of transmission attempts is reached.

3. THE H.264 VIDEO CODING STANDARD

We focus on the transmission of video data compressed according to the new ITU-T H.264 standard. The compression scheme follows the general structure of the ISO MPEG and ITU-T H.264 video coding standards, with some new features to achieve a higher compression efficiency. Some of them are briefly outlined in the following; refer to [8] and [13] for more details.

The base coding unit for transform coding is a 4x4 sample block. Macroblocks are composed of 16 luminance blocks and 4 blocks for each chrominance component. The transform coding is a separable integer transform with essentially the same properties of the traditional Discrete Cosine Transform (DCT). Regarding motion compensation, prediction using multiple reference frames is possible.

Consecutive macroblocks are grouped into a *slice*. The *slice* is important because it has the property to be independently decodable. This is useful to subdivide the coded stream into independently decodable packets, so that the loss of a packet does not affect the decoding of others (not considering the effects of motion compensation).

One of the most interesting characteristics of the H.264 standard is the attempt to decouple the coding aspects from the bitstream adaptation needed to transmit it over a particular channel. The part of the standard that deals with the coding aspects is called Video Coding Layer (VCL), while the other is the Network Adaptation Layer (NAL) [7]. One of the developed NAL is aimed to the problem of transporting data over an IP network using the Real-Time Transport Protocol (RTP) [3], which is well suited for real time multimedia transmissions.

Complete separation between the VCL and the NAL is, however, difficult to achieve. For instance, to improve error resilience, the VCL should create slices of about the same size of the packets handled by the NAL —which, in turn, should not split slices (a VCL entity) into different packets. Such cross-layer approach would benefit the error resilience of the transmission because all packets could be decoded independently.

In H.264 the subdivision of a frame into slices has not to be the same for each frame of the sequence; thus the decoder can flexibly decide how to slice each individual video frame. Slice should not be too short because that would cause a decrease of the compression ratio for two main reasons: the slice headers would reduce the available bandwidth, and the context-based entropy coding would become less efficient. Long slices, on the other hand, are more likely to contain transmission errors, which leads to reduced transmission efficiency and higher packet losses. In this chapter the performance trade-offs involved in the packet creation process will be investigated.

4. RESULTS

4.1 The simulation scenarios

Simulations have been carried out using the *ns* [15] network simulator fed with the well known *Foreman* video sequence. The video sequence is coded using the H.264 test model software [5], enabling most of the new characteristics of the H.264 standard, in particular multiple reference frames and Lagrangian optimized motion search for macroblocks down to 4×4 pixels size. The sequence size is CIF at 15 fps, and is encoded using a fixed quantization parameter, set to achieve a bit rate of about 256 kbit/s. The sequence length is 149 frames, and one B frame is introduced after each P frame. The transmitted sequence is obtained concatenating the base video sequence 80 times, reaching a length of 794.6 s at 15 fps. In order to improve error resilience, an I frame is interposed at the beginning of each repetition of the sequence (i.e., every 148 frames.) The video sequences are packetized according to the IP Network Adaptation Layer (NAL) specification of the H.264 standard and transmitted using the RTP/UDP/IP protocol stack. At the receiver, a playout buffer mechanism has been implemented to compensate the delay jitter of the packets. If not specified otherwise, the playout buffer size has been set to 1 s.

When present, the interfering data traffic is carried by greedy TCP connections; the NewReno version of TCP is used.

At the MAC layer, the duration of the time slot and of the DIPS time interval has been set to $20\ \mu s$ and $50\ \mu s$, respectively. When not specified otherwise, the maximum number of transmission attempts is set to 2.

The 802.11b radio channel is modeled as a Gilbert channel. Two states, *good* and *bad,* represent the state of the channel during an 802.11b time slot: an MPDU is received correctly if the channel is in state *good* for the whole duration of the MPDU transmission; it is received in error otherwise. We denote the transition probability from state *good* to state *bad* by p and the transition probability from state *bad* to state *good* by q. The average error probability, denoted by ϵ, is given by $p/(p+q)$; the average length of a burst of consecutive errors is equal to $1/q$ time slots. In the simulated scenarios the value of q is set to 0.9 so

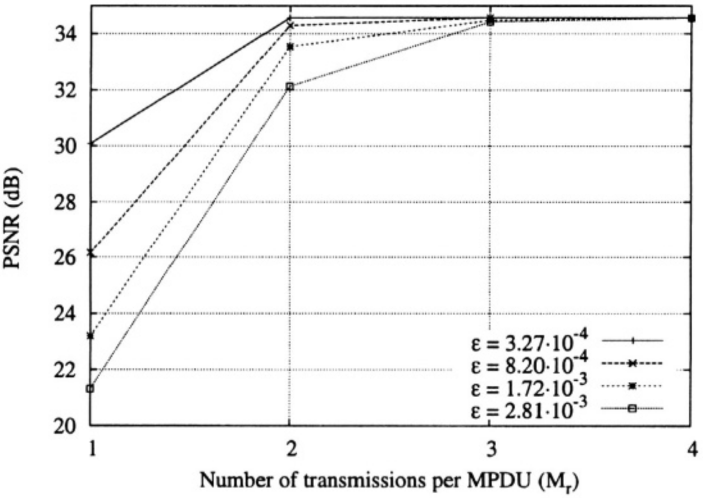


Figure 13-2. PSNR values as a function of the maximum number of transmission per MPDU at the MAC level. The results are plotted for various channel conditions.

that the average length of an error burst is equal to 1.1. The value of p varies so as to represent a range of channel conditions with different values of the average error probability.

We present two sets of simulation scenarios in a multi-service network which provides video transmission and data transfer. The first set of simulations analyzes the video transmission quality in a simple two-hop scenario (i.e., with one relay node only) under different network scenarios in which the error probability over the radio channel and the number of background data traffic sources vary. In the second set of simulations, we fix the number of background data sources and we investigate the impact of the number of relay nodes on the video quality.

4.2 Two-hop scenario

We first analyze the behavior of the transmission system when no background traffic is present in the network.

Figure 13-2 shows the impact of the maximum number of transmissions per MPDU on the perceptual video quality, measured by the peak SNR (PSNR), for four different channels. Almost optimal video quality can be achieved by setting the maximum number of transmission attempts per MPDU, M_r, to 3. If no retransmissions are present ($M_r = 1$), the quality rapidly decreases, showing that

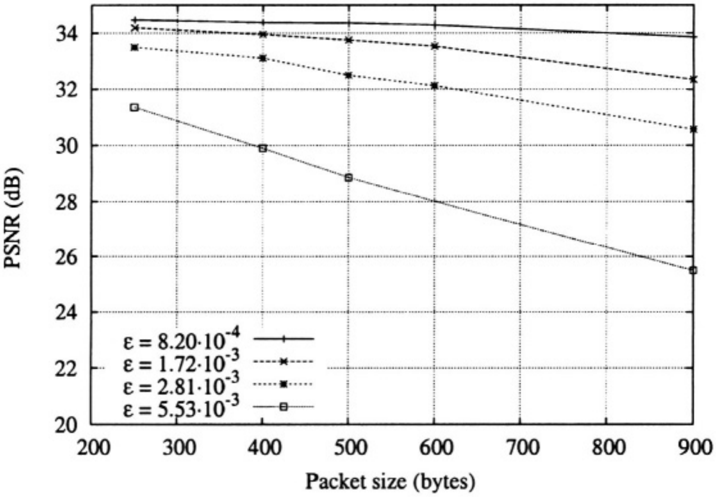


Figure 13-3. PSNR values as a function of mean packet size, for various channel conditions; M_r is equal to 2.

at least one retransmission at the MAC level is needed to obtain an acceptable video quality for realistic error probabilities.

Figure 13-3 shows the effect of the mean packet size on video quality, for various channel conditions, with M_r equal to 2. When the channel error probability is low ($\epsilon = 8.20 \cdot 10^{-4}$), the mean packet size has a limited influence on the video quality, thus larger video packets can be used, minimizing the MAC header overhead without incurring in an excessive quality degradation. For high error probabilities ($\epsilon = 5.53 \cdot 10^{-3}$), it is better to create smaller video packets during the encoding process so that the resulting packet error probability is minimized.

We now evaluate the impact that the following aspects of the system have on the quality of the video service: i) the interfering TCP traffic, ii) the error probability over the radio channel, iii) the setting at the MAC layer of the maximum number of allowed transmissions per MPDU.

Figure 13-4 shows the average delay perceived by UDP packets versus the slot error probability over the wireless channel when various numbers of interfering TCP sources are considered. The maximum number of transmissions per MPDU, M_r, is set to 2 in the MAC layer of the video traffic source. As the error probability over the wireless channel increases, the number of transmissions performed per MPDU increases so that longer times are needed to deliver

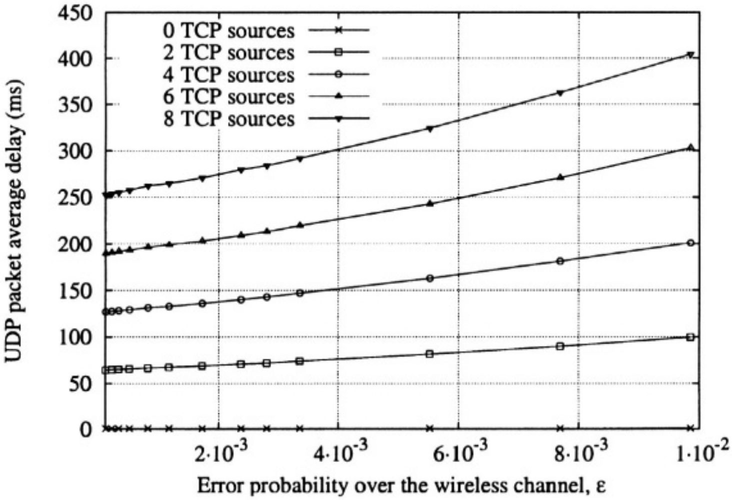


Figure 13-4. Average delay of UDP packets as a function of the error probability over the 802.11b wireless channel. The results are plotted for a varying number of TCP traffic sources and by setting the maximum number of transmission attempts for video traffic to be equal to 2.

the MPDUs and, thus, longer average delays are experienced at the UDP level. A similar behavior can be observed by letting the number of interfering TCP sources increase. The effect of a large number of interfering sources is twofold. On the one hand, it translates into a large collision probability which delays the access to the radio channel. On the other hand, when the channel is shared by a large number of sources, the channel capacity perceived by individual sources is smaller.

Let us focus on the case where no interfering TCP sources are considered. Depending on the radio channel conditions, the service time of video MPDUs is either equal to one or to two MPDU transmission times (remember that M_r is equal to 2). The service time results to be small enough that there is no queue at the MAC layer and the delay perceived by UDP packets is extremely small.

These results suggest some criteria for the choice of the set of services which can be provided by the system. Consider, for example, the case of interactive video services, for which an important QoS constraint consists in the average delay being kept lower than 150 ms. In this case, the number of interfering TCP sources should be limited. Up to 3 interfering sources are acceptable, while 4 sources can be admitted only if the average error probability over the radio

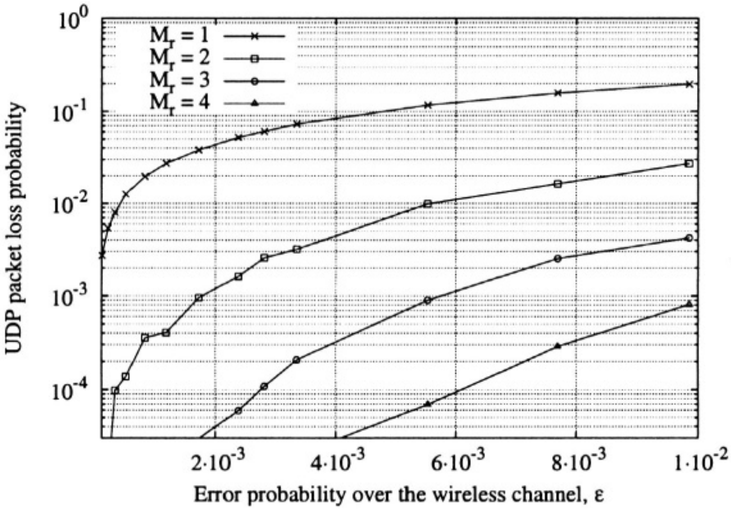


Figure 13-5. Loss probability of UDP packets as a function of the error probability over the 802.11b wireless channel, when no TCP traffic sources are considered. The results are plotted for different values of the maximum number of transmission attempts.

channel is small, say smaller than 0.003. More than 4 sources cannot be admitted even in the presence of very good channel conditions.

Figure 13-5 shows the impact of the radio channel conditions on the loss probability of UDP packets of the video stream when different values of the maximum number of transmissions per MPDU are considered. The curves show that some retransmissions are needed in order to keep the UDP packet loss probability to reasonable values. However, values of M_r as small as 3 are already enough to guarantee UDP loss probability smaller than 1%, even under bad channel conditions.

4.3 Impact of the number of hops

In this section, we investigate the impact of the number of hops, i.e., relay nodes, on the quality of service provided to the video and data services. As sketched in Figure 13-1, the video source shares with the TCP flows the path to the destination. We let the number of relay nodes increase from 1 (which corresponds to the previous scenario) to 4. Correspondingly, the number of hops in the path from the sources to the destination increases from 2 to 5. We set the number of TCP connections to 2.

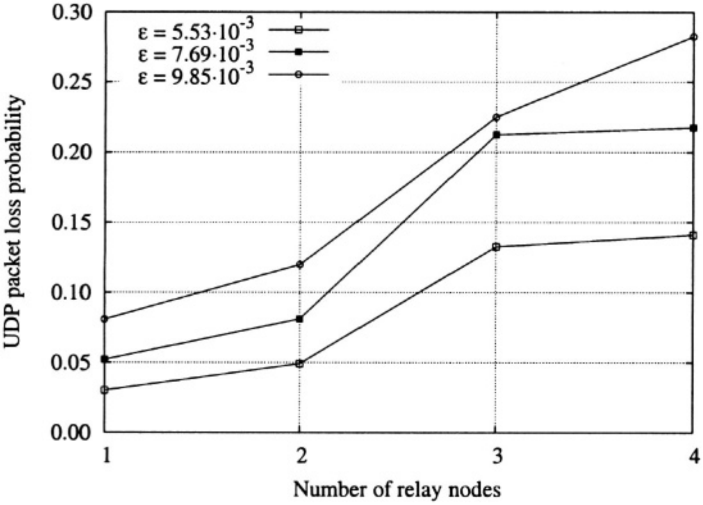


Figure 13-6. Loss probability of UDP packets as a function of the number of relay nodes, when two TCP traffic sources are considered. The results are plotted for different values of the error probability over the wireless channel.

In Figure 13-6, the UDP packet loss probability is plotted versus the number of relay nodes for three different values of the average error probability over the radio channel. As expected, the UDP loss probability increases with the number of relay nodes. The performance deterioration is limited when the number of relay nodes increases from 3 to 4 relay nodes, since the distance between the first and the last relay node is in this case large enough to allow for concurrent undisturbed transmissions.

The UDP packet average delay is shown in Figure 13-7 under the same scenario. The delay increase due to the higher number of hops makes it unfeasible to provide interactive video services (end-to-end delay smaller than 150 ms) with 4 relay nodes, while with 3 relay nodes that objective that can be achieved only under good channel conditions; i.e., when the error probability over the radio channel is lower than 0.5%.

As a more accurate measure of the quality of service perceived by video users, the PSNR is plotted in Figure 13-8 versus the number of relay nodes for different channel conditions; the playout buffer is set to 1 s, which represents a suitable value for typical streaming scenarios. The quality decreases with the error probability and acceptable service can be provided only under very good channel conditions. Clearly, as the delay constraint tightens, the difficulty to provide

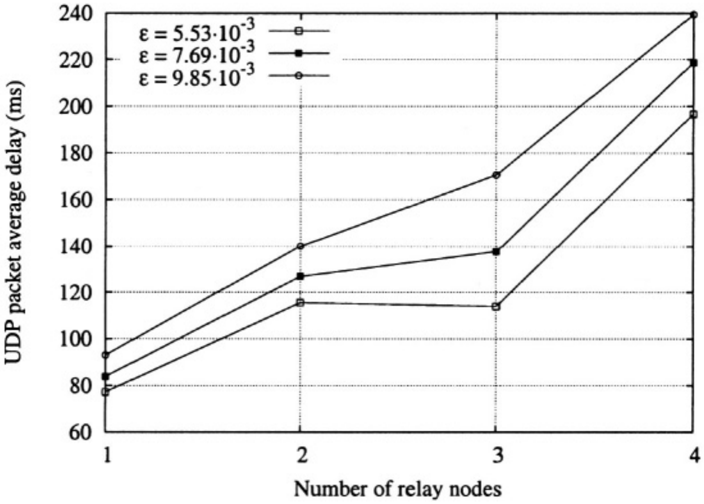


Figure 13-7. Average delay of UDP packets as a function of the number of relay nodes, when two TCP traffic sources are considered. The results are plotted for different values of the error probability over the wireless channel.

acceptable quality of service to the video users increases. In Figure 13-9 the PSNR is plotted versus the number of relay nodes for different values of the playout buffer under average error probability equal to $5.53 \cdot 10^{-3}$. Even if the channel is good, tight delay constraints are difficult to meet when the number of hops is large, say larger than 3. For example, very low quality is provided with a constraint of 150 ms when only 3 hops are needed to reach the destination. These results suggest that, in a multi-hop network scenario, high-quality real-time video can be provided only under loose delay constraints and limited distance (in terms of number of hops) between source and destination.

Finally, we assess the impact of the number of relay nodes on the performance of data traffic. The throughput achieved by TCP connections is shown in Figure 13-10. The increase of both loss probability and delay makes the TCP throughout drastically decrease with the number of relay nodes.

5. CONCLUSIONS

The behavior of H.264-coded video transmission over a wireless 802.11b ad hoc network scenario has been analyzed. The influence of some of the main parameters involved in the transmission system has been studied by means of

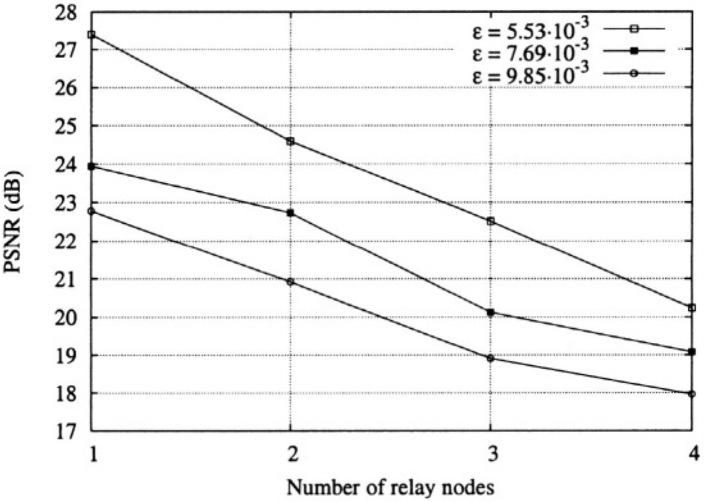


Figure 13-8. PSNR values as a function of the number of relay nodes, when two TCP traffic sources are considered and the playout buffer is equal to 1 s. The results are plotted for different values of the error probability over the wireless channel.

network simulations. Various scenarios have been tested, with different levels of background interfering traffic and with different network configurations. Results give a clear indication on how to select the system parameters. In particular, we have observed that a video packet size as small as 300 bytes should be used when channel conditions are not favorable. A maximum number of transmission attempts at the MAC layer equal to 3 enables us to obtain a high PSNR for most channel conditions. Moreover, in case of interactive video services, the number of TCP sources that can be admitted in the network should be limited and the number of hops must be kept very small, i.e., smaller than 4, in order to meet the QoS requirements.

References

[1] (1999). Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ISO/IEC 8802-11, ANSI/IEEE Std 802.11.*

[2] Freytes, M., Rodriguez, C.E., and Marques, C.A. (2001). Real-Time H.263+ Video Transmission on 802.11 Wireless LANs. In *Int. Conf. on Information Technology: Coding and Computing*, Las Vegas, NV.

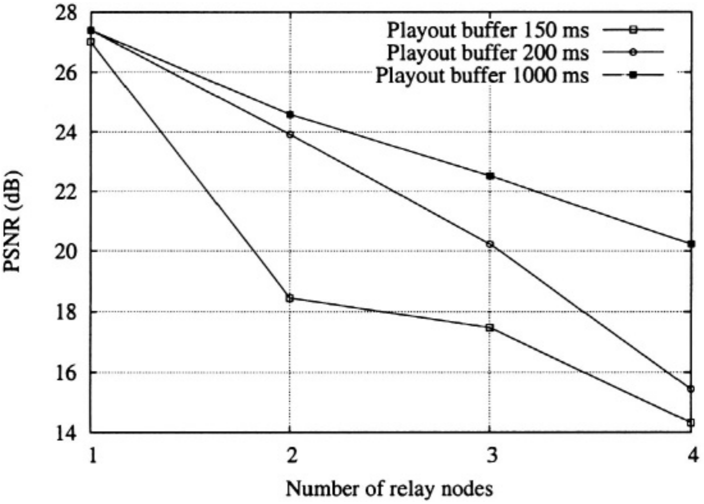


Figure 13-9. PSNR values as a function of the number of relay nodes, when two TCP traffic sources are considered and the error probability over the wireless channel is equal to $5.53 \cdot 10^{-3}$. The results are plotted for different values of the playout buffer.

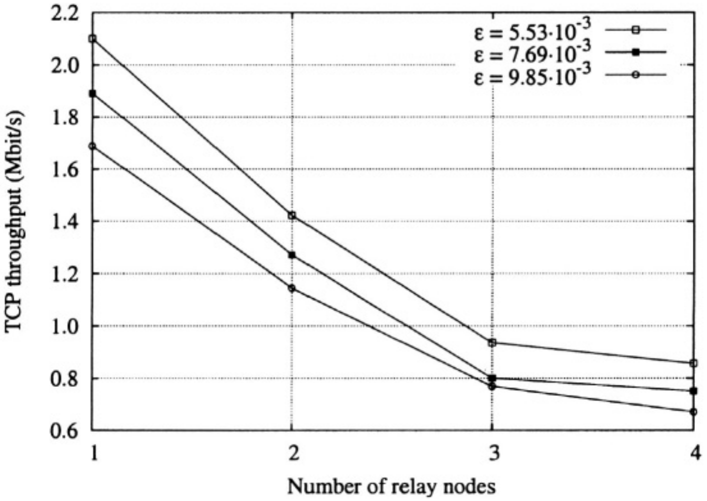


Figure 13-10. Throughput of the TCP sources as a function of the number of relay nodes. The results are plotted for different values of the error probability over the wireless channel.

[3] H. Schulzrinne, S. Casner, R. Frederick and Jacobson, V. (1996). RTP: A Transport Protocol for Real-Time Applications. *RFC 1889.*

[4] Hu, Y.-C. and Johnson, D.B. (2002). Design and demonstration of live audio and video over multihop wireless ad hoc networks. In *IEEE MILCOM 2002,* pages 1211–16.

[5] ITU-T VCEG (2002). Test Model Long Term (TML) 9.7. *URL: ftp://standard.pictel.com/video-site.*

[6] Johnson, David B and Maltz, David A (1996). Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing,* volume 353. Kluwer Academic Publishers.

[7] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, no. 7, pp. 645–656, July 2003.

[8] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced video coding for generic audiovisual services," *ITU-T,* May 2003.

[9] Lin, S., Wang, Y., Mao, S., and Panwar, S. (2002). Video transport over ad-hoc networks using multiple paths. In *IEEE International Symposium on Circuits and Systems (ISCAS),* pages 57–60.

[10] Majumdar, A., Sachs, D. G., Kozintsev, I.V., Ramchandran, K., and Yeung, M.M. (2002). Multicast and Unicast Real-Time Video Streaming Over Wireless LANs. *IEEE Transactions on Circuits and Systems for Video Technology,* 12(6):524–534.

[11] Mao, S., Lin, S., Panwar, S.S., and Wang, Y. (2001). Reliable transmission of video over ad-hoc networks using automatic repeat request and multipath transport. In *54th IEEE Vehicular Technology Conference (VTC) 2001 Fall,* pages 615–19.

[12] Perkins, C.E. and Royer, E.M. (1999). Ad-hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99),* pages 90–100.

[13] T. Wiegand, G. J. Sullivan, G. Bjfintegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, no. 7, pp. 560–576, July 2003.

[14] Suzuki, T. and Tasaka, S. (1999). Performance Evaluation of Integrated Video and Data Transmission with the IEEE 802.11 Standard MAC Protocol. In *Proceedings ofGLOBECOMM,* volume 1b, pages 580–586.

[15] UCB/LBNL/VINT (1997). Network Simulator – ns – version 2. *URL: http://www.isi.edu/nsnam/ns.*

This page intentionally left blank

Chapter 14

DWT IMAGE COMPRESSION FOR MOBILE COMMUNICATION

Lifeng Zhang, Tahaharu Kouda, Hiroshi Kondo, Teruo Shimomura
Kyushu Institute of Technology, Sensui-cho, tobata-ku, Kitakyushu, Japan 804-8550
Email:zhaing@elcs.kyutech.ac.jp

Abstract: DWT image compression for mobile communication is presented. Discrete wavelet transform (DWT) with Haar mother function is utilized in this paper. The exact location information of the important DWT coefficients is generally needed for reconstructing the image. In this work, however, such information is not needed because it can be obtained from the DWT- approximation. Through a one dimensional directional difference operator not only the exact location information of the DWT coefficients but also the rough estimate of the coefficient itself can be obtained from the DWT-approximation when Haar mother wavelet function is utilized. The direction of the difference operator are different each other according to the DWT-details (horizontal, vertical and diagonal detail). This paper shows highly efficient image compression can be achieved when such DWT-approximation information is utilized well.

Keywords: Image compression, wireless communication, Haar transform, Discrete wavelet transform

1. INTRODUCTION

In these years mobile communication is broadly extended in variety ways. For such mobile communication technology an image compression technique is very important for fast transmission of an image. The world wide standard of compression coding image is JPEG and/or JPEG 2000

[1,2,3,4,5]. They are excellent but still it is needed to make effort for the higher efficiency in image coding. The presented method is lossy compression one and completely compatible with such a traditional coding method. Using a directional difference on the DWT-approximation we can get a rough estimate of the DWT-detail in the case that Haar mother wavelet function is employed. A higher image compression rate is attained if the above nature is utilized well.

2. DISCRETE WAVELET TRANSFORM

Discrete wavelet transform (DWT) has an interesting nature for image compression. DWT is a relatively new transform and many mother wavelet functions are there. Hence we can choose an appropriate mother function for the problem under consideration. Here we utilize a Haar function as a mother wavelet because the resulting DWT coefficients are similar to the difference image. The DWT is defined as (Transform):

$$S_k^{(J)} = \sum_n \overline{P_{n-2k}} S_n^{(J-1)} \tag{1}$$

$$\omega_k^{(J)} = \sum_n \overline{q_{n-2k}} S_n^{(J-1)} \tag{2}$$

where $S_k^{(J)}$ is a scaling coefficient as

$$S_k^{(J)} = \int_{-\infty}^{\infty} f(t)\overline{\varphi_{J,k}(t)}dt \quad (\varphi_{J,k}(t): \text{scaling function}) \tag{3}$$

and $\omega_k^{(J)}$ is a wavelet coefficient as

$$\omega_k^{(J)} = \int_{-\infty}^{\infty} f(t)\overline{\Psi_{J,k}(t)}dt \quad (\Psi_{J,k}(t): \text{ wavelet function}) \tag{4}$$

P_k is a scaling function sequence of Doubchies. q_k is a wavelet function sequence and has a relation with P_k as

$$q_k = (-1)^k P - k \tag{5}$$

The inverse DWT is expressed as

$$S_n^{(J-1)} = \sum_k \left[P_{n-2k} S_k^{(J)} + q_{n-2k} \omega_k^{(J)} \right]$$
$$(S_k^{(0)} = f(n):\ \text{original signal}) \tag{6}$$

From these equations we can see that DWT coefficients are calculated by a recursive equation. It means the calculation is fast and simple. Especially in the case where Haar function is utilized as a mother wavelet one the calculation is the simplest and furthermore it has an interesting nature expressed in the next section.

3. IMAGE COMPRESSION

3.1 Haar wavelet function

The Haar function has a simple structure shown in Eq. (7):

$$\psi(t) = \begin{cases} 1\ (0 \le t < 1/2 \\ -1 (1/2 \le \mathrm{t} < 1) \\ 0\ (\text{otherwise}) \end{cases} \tag{7}$$

Fig. 14-1 shows a two dimensional DWT with a Haar mother wavelet function.

The left upper quadrant is called a level-1 DWT approximation. This looks very similar to the original image but the size is a half of the original one through a down sampling. The right upper, the left lower and the right lower quadrant is called the vertical, the horizontal and the diagonal detail respectively. The vertical detail includes the information about a vertical element of the original image. Then it is concerned with a horizontal directional (one dimensional) difference of the original image. Actually the horizontal directional difference image of the DWT approximation is similar to the vertical detail.

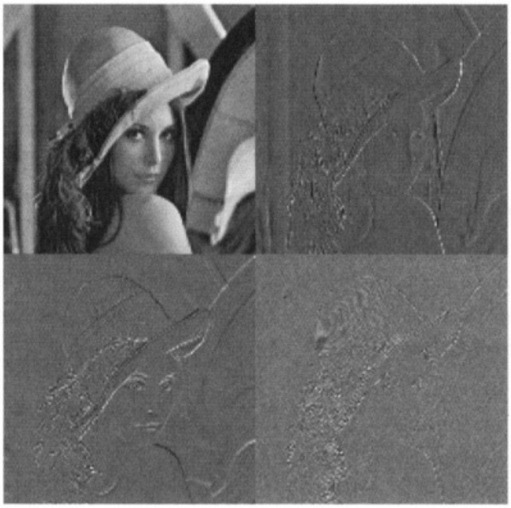


Figure 14-1. DWT (level-1)

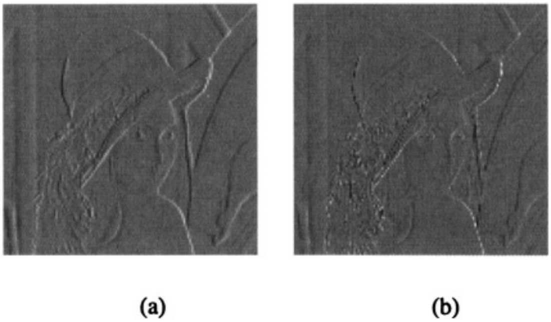


Figure 14-2. (a) Difference image (Horizontal), (b) Vertical DWT-detail

Fig. 14-2(a) and 14-2(b) are the horizontal directional difference image of the DWT approximation and the vertical DWT detail respectively. And in fact the cross correlation coefficient between these two images is relatively large (in the above image it is 0.735). Similarly the horizontal DWT detail looks like the vertical directional (one dimensional) difference image of the

DWT approximation. Fig. 14-3(a) and 14-3(b) show the similarity. The diagonal DWT detail corresponds to the diagonal directional difference image. In this case, first we take a horizontal directional difference and then for the resulting result we take a vertical difference. Fig. 14-4(a) and 14-4(b) show the relation between these two images.

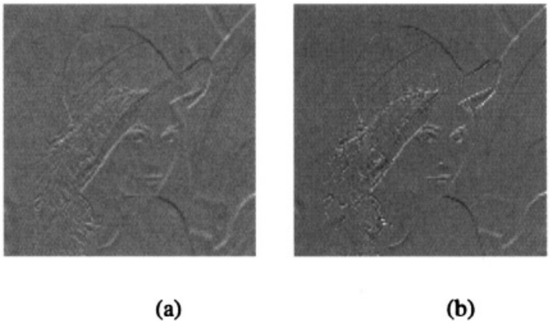


Figure 14-3. (a) Difference image (Vertical), (b) Horizontal DWT-detail

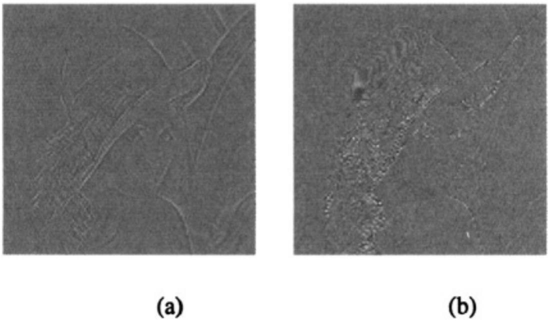


Figure 14-4. (a) Difference image (Diagonal direction), (b) Diagonal DWT-detail

3.2 Reconstruction

As described above each directional difference image is very similar to the corresponding detail. Actually replacing these DWT details by the corresponding differences we get Fig. 14-5 through the inverse DWT. The SNR of Fig. 14-5(a) is 27.3 dB. And that of Fig. 14-5(b) is 28.5 dB.

In this case it is better to take edge enhancement of the DWT-approximation image. There are many cases such a picture quality as that of

Fig. 14-5 is enough. For example such images can be utilized well for a mobile phone. Then at the receiver side we can reconstruct the images whose picture quality is like those of Fig. 14-5(a) and 14-5(b) when we can get only the DWT approximation. This means that the data to be sent is just 1/4 of the original. The DWT approximation can be compressed and coded through an ordinary JPEG or JPEG2000 method. Hence, for example, if the JPEG method gives us 4/5 reductions of the image data then the needed data to be transmitted is only 1/20 of the whole image data.

3.3 Image Compression

Many DWT coefficients are nearly zero but actually not equal to zero. Consequently we had better take the above corresponding difference instead of zero values. Then taking a threshold value processing the bit rate reduction can be achieved:

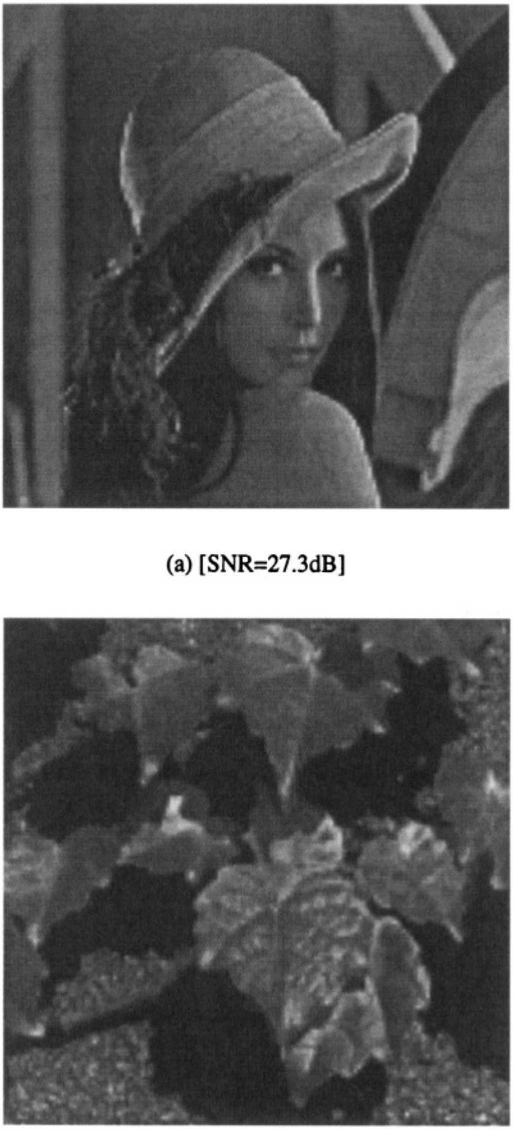
$$\hat{D}_{dwT}(i,j) = \begin{cases} D(i,j) & : |D(i,j)| \le \gamma \\ D(i,j) + \varepsilon(i,j) & : | D(i,j)| > \gamma \end{cases} \tag{8}$$

(γ: Threshold value)

where,

$$\varepsilon(i,j) = D_{dwT}(i,j) - D(i,j) \tag{9}$$

and $D_{dwT}(i,j)$ is DWT detail, $\hat{D}_{dwT}(i,j)$ is the estimate of $D_{dwT}(i,j)$, *D(i,j)* is the corresponding difference described in the former section, and ε is the error of $\hat{D}_{dwT}(i,j)$ from the true value $D_{dwT}(i,j)$. $\varepsilon(i,j)$ has a zero mean and small variance. Since we can create *D(i,j)* from the DWT approximation at the receiver side we can reconstruct the image if we obtain $\varepsilon(i,j)$. Then $\varepsilon(i,j)$ must be transmitted to the receiver side. Consequently only fewer bits are needed to transmit because $\varepsilon(i,j)$ has small variance with zero mean.



(a) [SNR=27.3dB]

(b) [SNR=28.5dB]

Figure 14-5. Reconstructed image with no details

4. SIMULATIONS

If we only need almost 30 dB picture quality for the reconstructed image then we can use a directional difference value described above and we do not need information almost DWT-details at all. That is, the image reconstruction is performed through the DWT-approximation only. A higher picture quality, however, when we need, DWT-detail information is needed. Figures 14-6 and 14-7 are the reconstructed image with an average bit rate 0.15 bits/pel for the DWT-details. The SNRs of these are 34 dB (lenna) and 35 dB (Happa) and these total average bit rates are both 0.47 bits/pel. The DWT-approximation is coded by using an ordinary JPEG method and its average bit rate is 1.28 bits/pel. The picture qualities of these reconstructed images are about the same as those of JPEG 2000 and look excellent for a practical use.

5. CONCLUSION

A new image compression method using DWT has been presented. In this work DWT-details are estimated through its DWT-approximation. The directional one-dimensional difference is introduced for this estimation. Horizontal, vertical, and diagonal DWT-details have the different corresponding one-dimensional differences respectively. In case of Haar mother wavelet these directional one-dimensional difference is worthwhile. We utilize our presented method only for level one-DWT of an image. It looks interesting to use the method also for level two-DWT. It will be considered as a future problem. In any case making IC and/or LSI with the presented method is easy because of Haar wavelet and one dimensional difference. Hence applying this method to a mobile phone is one of the best utilities.

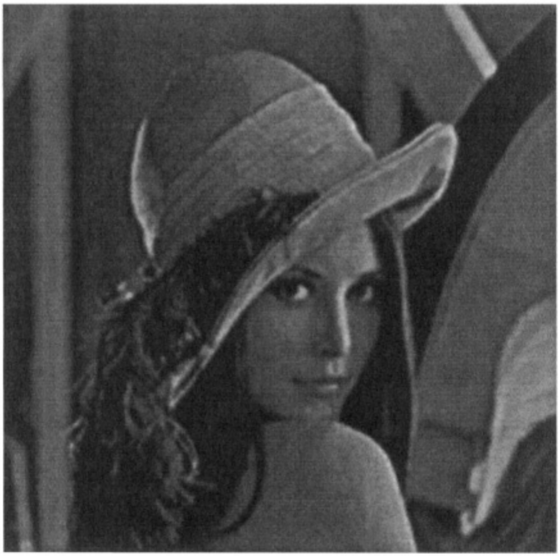


Figure 14-6. Reconstructed image [SNR=34dB]

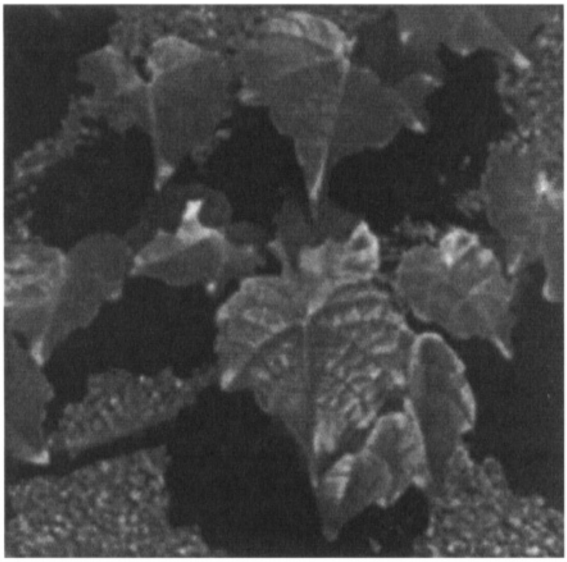


Figure 14-7. Reconstructed image [SNR=35dB]

REFERENCES

[1] ISO/IEC 15444-1, Information technology-JPEG 2000 image coding system-part1: Core coding system, ISO/IEC JTC 1/SC 29/WG1, Jan 2001.

[2] ISO/IEC JTC/1/SC29/WG1 N1987, Motion JPEG 2000 Committee Draft 1.0, Dec. 2000.

[3] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," Applied and Computational Harmonic Analysis (ACHA), vol. 5, no. 3, pp. 332–369, 1998.

[4] D. Taubmam, "High performance scalable image compression with ebcot," IEEE Trans. on Image Processing, vol. 9, no. 7, pp. 1158–1170, July 2000.

[5] Wenjun Zeng, Scott Daly, and S. Lei, "Point-wise extended visual masking forjpeg-2000 image compression," IEEE ICIP, Vancouver, Canada, Sept. 2000.

Chapter 15

LINK-ADAPTIVE VARIABLE BIT-RATE SPEECH TRANSMISSION OVER 802.11 WIRELESS LANS *

Antonio Servetti[1] and Juan Carlos De Martin[2]
[1] *Dipartimento di Automatica e Informatica;* [2]*IEIIT-CNR Politecnico di Torino, Corso Duca degli Abruzzi, 24 —I-10129 Torino, Italy Email: demartin@polito.it*

Abstract We present an adaptive technique to transmit speech over 802.11 wireless packet networks. According to the proposed scheme, the speech coding rate of a network-driven variable bit-rate coder is selected to match the instantaneous wireless channel conditions: higher rates (i.e., larger packets) for low error rates, lower rates (i.e., smaller packets) when the channel is noisy. Packet size is, in fact, directly related to the probability of retransmission, one of the major sources of delay and losses in contention-based medium access control. Network simulations using the 3GPP GSM-AMR speech coding standard show that the adaptive approach can address the stringent quality of service requirements for two-way interactive speech applications over wireless packet networks, reducing packet loss rates and end-to-end delays.

Keywords: wireless network, 802.11, voice over IP, adaptive speech, GSM AMR, link adaptation, adaptive packet size

1. INTRODUCTION

Wireless technology keeps changing the communications scenario. Wireless local area networks (WLAN's), in particular, are being enthusiastically adopted by users worldwide, shaping a new world where tetherless access will be possible not only in homes and offices, but also in an increasing number of previ-

*This work was supported in part by CERCOM, the Center for Wireless Multimedia Communications, Torino, Italy, http://www.cercom.polito.it.

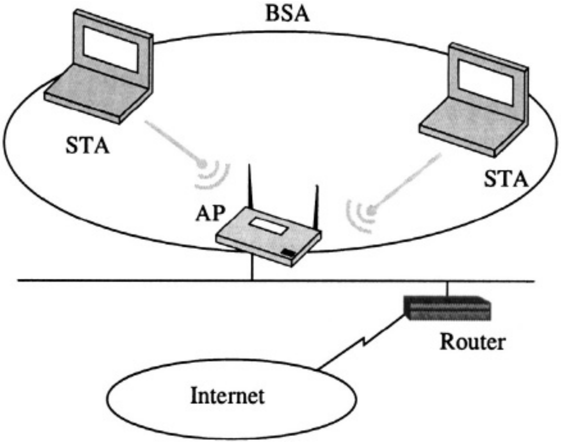


Figure 15-1. 802.11-based network communications scenario.

ously unconnected places, like shopping malls, libraries, trains and other means of mass transportation, even private motor vehicles. As soon as seamless integration with wide-area coverage provided by 2.5G/3G cellular wireless infrastructures is reached, wireless access will likely become the most common form of network access for an increasing number of users.

The IEEE 802.11 WLAN standard, based on the definition of the medium access control (MAC) protocol and the physical layer (PHY) specifications, became available in 1999 [1] and since then has emerged as the most successful and most widely deployed WLAN standard. Figure 15-1 shows a simple 802.11-based network scenario, with two mobile stations and an access point (AP) connected to a wired LAN.

So far, the main usage of Wireless LAN's has been limited to Internet based services like Web browsing, e-mail, and file transfers. However, as already happened in the traditional wired LAN's, a strong interest is emerging towards multimedia applications over WLAN's. Among them, interactive voice communication looks particularly appealing. WLAN-based telephony, in fact, could not only replace and significantly extend traditional cordless telephony, but also compete with cellular telephony in at least a certain number of scenarios. Moreover, such technology would have all the typical Voice over IP advantages, including a single infrastructure for both data and voice, greater flexibility with respect to traditional telephony, and the possibility of introducing new value-added services.

Several challenges, however, need to be addressed to make WLAN telephony as successful as cellular and wired telephony. Not only the actually available bandwidth for WLAN's can be significantly below that of their wired counterparts, but wireless links are also strongly time-varying and may have high error rates. Other issues are specific of 802.11 WLAN's, including the MAC layer effects on performance, the consequences of interfering data traffic, and determining the optimal configurations for both Access-Point-based and ad-hoc 802.11 networks for a given application scenario.

Previous research evaluated the performance of interactive voice traffic over Wireless LAN's [7][19], mainly by means of statistical analysis of throughput and packet losses to assess the number of supportable voice conversations.

In this chapter we present a new technique for improving the quality of interactive voice communications over 802.11 wireless packet networks. The operating rate of a network-driven variable-bitrate speech coder is chosen on a frame-by-frame basis according to the instantaneous channel conditions: higher rates (i.e., longer packets) when the channel is good, lower rates (i.e., shorter packets) when the channel is poor. Performance is measured in terms of average packet losses and average delay, with and without interfering traffic, using a network simulator. The disadvantage of temporarily lowering the source coding quality is clearly offset by the advantages deriving from lower packet losses and delays. The proposed system thus consistently outperforms constant-bitrate speech transmission at the same average bit-rate.

The chapter is organized as follows. In Section 2, we introduce the wireless Voice over IP scenario and the adaptive multi-rate speech coder. In Section 3, we describe the proposed speech transmission scheme. Results and conclusions are presented in Section 4 and 5, respectively.

2. VOICE OVER 802.11 WLAN'S

Voice over IP over wireless packet networks is becoming increasingly attractive. In particular, the widespread adoption of WLAN technology is creating the basis for the introduction of a new form of cordless telephony in offices, homes, hospitals, etc.

Two-way conversational applications, however, are characterized by stringent requirements on the end-to-end delay. The upper limit for one-way delay is set to only 150 ms, according to the guidelines of ITU-T Recommendation G.114. Moreover, packet losses should be kept below 1% to prevent significant perceptual degradation.

The WLAN environment is quite challenging on two counts: the wireless link is inherently noisy, due to fading and interference; the contention-based medium access control (MAC) layer and the retransmission-based error-control scheme may introduce strong delays as well as packet losses.

Efficient WLAN-based telephony systems must thus be designed carefully to overcome the difficulties of the environment if toll quality service is to be delivered.

2.1 IEEE 802.11 Wireless LAN's

Users may conveniently access the Internet via Wireless LAN technology. Bridging functionality is provided by access points that interconnect wireless nodes to the wired infrastructure, i.e. IEEE 802.11 networks in infrastructure mode. The IEEE 802.11b physical layer —which operates in the license free 2.4 GHz ISM (Industrial, Scientific, and Medical) band— implements a Direct Sequence Spread Spectrum (DSSS) system with an 11 Mbps top bit-rate. The MAC sublayer is responsible for the channel allocation procedures, frame formatting, error checking, fragmentation and reassembly. The fundamental transmission medium defined to support asynchronous data transfer on a best effort basis is called Distributed Coordination Function (DCF). It operates in a contention mode requiring all stations to contend for access to the channel for each packet transmitted. Contention services promote fair access to the channel for all stations.

In the IEEE 802.11 MAC, each data-type frame consists of the following basic components: a MAC header, a variable length information frame body, and a frame check sequence. All fields except the frame body (28 bytes in total) contribute to the MAC protocol data unit (MPDU) overhead for a data frame. Upon packet transmission the destination station positively acknowledges each successfully received packet by sending an ACK frame back to the source station. When an ACK is not received, the source station contends again for the channel to transmit the unacknowledged packet and, in case of further errors, retries until a maximum retry limit is reached.

2.2 The GSM AMR Speech Coding Standard

The GSM Adaptive Multi-Rate (AMR) standard [10] is a state-of-the-art network-driven variable-bitrate speech coder. Its operating bit-rate can be chosen on a frame-by-frame basis to match the instantaneous channel conditions. In the case of cellular telephony, the objective is to change the ratio between bandwidth devoted to speech and bandwidth devoted to forward error correction. For the proposed technique, the objective is to use the speech rate, i.e. the speech packet size, most suitable for any given 802.11 channel condition.

The GSM-AMR speech coder is a multi-rate ACELP coder with 8 modes operating at bit-rates from 12.2 kbps to 4.75 kbps. The coder modes are integrated in a common structure, where the bit-rate scalability is obtained by adjusting the quantization schemes for the different parameters. The frame size is 20 ms, consisting of 4 sub-frames of 5 ms each.

3. ADAPTIVE REAL-TIME MULTIMEDIA TRANSMISSION

The time-varying nature of wireless channels as well as network congestions may significantly degrade the quality of speech communications. Adaptive transmission techniques are designed to match the time-varying nature of the wireless channel, thus typically delivering the desired level of QoS for real-time multimedia more effectively than non-adaptive schemes (see, e.g., [12]).

Although non-adaptive schemes tend to be simpler and potentially more robust, they are optimal only for the operating point used for their design: their performance quickly decreases when the scenario worsens and they also cannot exploit better conditions when available. Marginal channel conditions are quite common in real systems: these are encountered, for example, just prior to handoff or during deep shadowing, as when a mobile station suddenly goes behind a building or a hill.

As the condition of the radio channel varies with both the location and mobility of the terminal, it is therefore desirable to employ an adaptive solution providing the user with maximum quality for any given channel condition. Adaptive solutions have been proposed in the literature for different layers of the network infrastructure and the next section will present a brief survey to the reader.

3.1 Adaptive Techniques for 802.11

Previous works presented several techniques —located at different network layers— to increase the quality of service of wireless networks by means of channel or traffic adaptation.

The IEEE 802.11b physical (PHY) layer provides four PHY rates from 1 to 11 Mbps at the 2.4 Ghz band. Link adaptation mechanisms have been discussed so that the proper PHY rate can be adaptively selected to combat the variation of the wireless medium condition, hence improving the goodput performance of a WLAN [14]. In fact the higher the PHY rate, the shorter the transmission time in one transmission attempt, but the more likely that the transmission will fail, thus engendering retransmissions.

A link-level adaptation approach can be adopted to change the transmitted packet size according to variations of the channel quality [6], since shorter packets are more suitable for noisy channel conditions. Transmission robustness can then be increased enabling fragmentation whenever a local station estimator (on the transmitter side) evaluates the channel error rate to be above a given threshold.

Besides channel noise, also network congestions and interference can threaten the performance of real-time multimedia communications over mobile networks. Application level adaptation based on packet loss and delay reports from the receiver can be an effective approach to reduce source rate and packet size to match the available bandwidth [5][13].

The solutions presented so far can provide better QoS, but they still suffer from some limitations. When physical transmission rate is reduced also transmission speed and throughput decrease. If fragmentation is used when the channel is noisy, the overhead introduced by fragment headers can cause even more congestion. Source rate adaptation at the application level can be too slow to face the time varying error characteristic of the wireless channel.

3.2 Link-Adaptive 802.11 VBR Speech Transmission

A new technique is presented whose purpose is to introduce link-layer packet size adaptation by means of adaptive-rate speech coding.

In IEEE 802.11 wireless LAN's, packets received without errors are positively acknowledged by the sender, otherwise they are retransmitted until a given maximum number of retransmissions is reacherd. Changing the transmitted packet size according to the error rate of the wireless link between the mobile terminal and the access point can reduce the number of retransmissions needed to successfully send a packet. Smaller packets, in fact, are less likely corrupted than larger ones, and they are, therefore, more suitable for noisy channels, at the cost of increased overhead. Less retransmissions result into less packet losses, lower end-to-end delay and less channel congestion.

In the case of speech communications, compressed speech should be transmitted using larger packets when the wireless channel is good, and smaller packets when the channel is poor. Speech coded using PCM-based techniques, e.g., the ITU-T G.711 64 kb/s coding standard, is particularly suitable in this regard, since it can be packetized very flexibly. Most modern, bandwidth-efficient speech coders, however, typically code 10–20 ms speech segments into frames of fixed size. In this case, packet sizes can be varied by changing the amount of speech data (e.g., frames) placed into a packet: several speech frames could be packetized together when the channel is good, just a frame or two when the channel is noisy. Changing, however, the amount of speech data —be that PCM samples or frames— encapsulated into packets causes causes potentially high delays; moreover, such delay would be time-varying.

A different, constant-delay approach based on network-driven variable bit-rate speech coders is possible. Variable bit-rate coders can compress speech segments in frames of different dimensions according to the selected codec mode. For example, the GSM AMR can generate —as discussed in Section 15.2.2— eight different frame sizes, all representing 20 ms of speech, ranging from 95 bits (lowest quality) up to 244 bits (highest quality). When a variable-rate speech codec is available, higher rates (i.e., larger packets) can be used for good channel conditions, lower rates for noisy conditions. The trade-off is, therefore, between source coding quality —which is proportional to the speech coding rate— and transmission performance —which depends on the packet size. For at least some scenarios, the expectation is that the source coding degradation experienced when the channel is noisy (when lower coding rates are employed) is more than compensated by better transmission performance in terms of lower packet losses, end-to-end delay and network congestion.

The proposed technique requires a channel estimation algorithm to select the optimal output rate of the speech coder at any given time instant. The GSM AMR standard leaves link quality estimation open. However, it provides an example solution, which is based on burst-wise C/I estimates [11]. Channel estimation schemes are covered in some detail in [4]. A carrier signal estimate is computed using a training sequence known a priori. By comparing the received training sequence and the known training sequence the receiver can estimate the current C/I and communicate it to the sender. For codec mode adaptation, the measure of the instantaneous channel quality has to be mapped to codec modes. This is in principle done by quantizing the measurement where the levels of the quantizer are mapped onto the different codec modes.

For the proposed technique, channel quality measurements are roughly quantized in two states that represent good and bad channel conditions. In the bad state, large packets have a higher probability to be in error and, therefore, to be retransmitted, while small packets are more easily received without errors.

3.3 Wireless Transmission Model

Wireless transmission is error prone with non-stationary error characteristics. Bit error rates as bad as 10^{-2} or 10^{-3} are reported [3]. They are caused by path loss, fast fading due to movement and multipath propagation, slow fading due to moving beyond large obstacles, noise and interference from other networks or devices like microwave ovens.

A widely used model for the error characteristics of a wireless channel is the Gilbert-Elliot two state Markov model [20][21][8], where each state represents a Binary Symmetric Channel (BSC). Each state is assigned a specific constant Bit Error Rate (BER): in the "good" state (G) errors occur with low probability p_G, while in the "bad" state (B) they happen with high probability p_B ($p_B \gg p_G$). Within one state errors are assumed to occur independently from each other. For a complete specification of the model the values p_{GB} and p_{BG}, that represent the probability to switch from the good state to the bad state and vice versa, are necessary and sufficient.

The steady state probabilities of being in the states G and B are

$$\pi_G = \frac{p_{BG}}{p_{BG} + p_{GB}}, \pi_B = \frac{p_{GB}}{p_{BG} + p_{GB}}, \tag{1}$$

respectively. Hence the average bit error rate produced by the Gilbert-Elliot channel is

$$p = p_G \pi_G + p_B \pi_B. \tag{2}$$

As established by measurements, BER's of the order of 10^{-5} and 10^{-3} are typical for the good and bad state, respectively. For simplicity we assume that state transitions occur only at multiples of 20 ms and that a packet is entirely sent in one of the two states (no state transitions occur in the middle of a packet.)

Given these restrictions we can express the packet error probability as a function of the packet size: with a b-bit packet and a BER of p, the packet will be considered corrupted and therefore discarded with probability

$$1 - (1 - p)^b. \tag{3}$$

For a Gilbert-Elliot channel the packet error rate can then be expressed as

$$p_{pckt} = \pi_G[1 - (1 - p_G)^{b_G}] + \pi_B[1 - (1 - p_B)^{b_B}], \tag{4}$$

where b_G and b_B are the packet size (in bit) during the channel good and bad states, respectively.

If retransmissions are allowed at the MAC level the source station can transmit a packet at most N times before discarding it. The perceived correct rate at

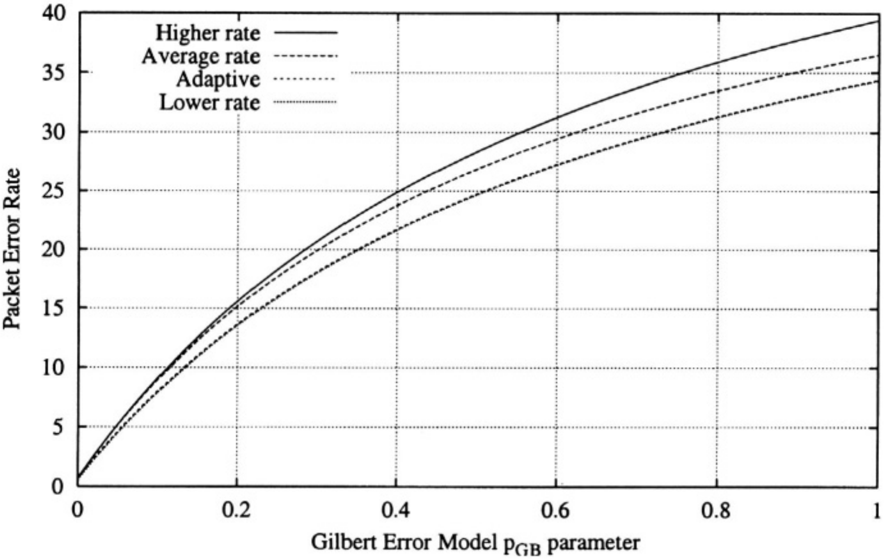


Figure 15-2. Packet loss rate as a function of the Gilbert-Elliot model parameter p_{BG}. The value p_{GB} is fixed at 0.6. p_G is set to 1×10^{-5} and p_B to 1.5×10^{-3}. The packet size at the physical level for the higher and lower rates are 704 and 560 bits respectively. The average rate transmission has a constant packet size equal to the average size of the adaptive transmission packets.

the transport protocol layer is:

$$P_{pckt}(correct) = \sum_{i=1}^{N}(1 - p_{pckt})p_{pckt}^{i-1} = 1 - p_{pckt}^{N}, \qquad (5)$$

where N is the maximum number of transmission at MAC (DCF mode) and p_{pckt} is the packet loss rate at physical layer. Consequently, the perceived loss rate at transport protocol is

$$P_{pckt} = p_{pckt}^{N}. \qquad (6)$$

In Figure 15-2 the packet loss rate at the transport layer is plotted for different packet sizes corresponding to different speech coding rates. Three constant packet sizes are considered: a 12.2 kb/s speech coding rate ($rate_A$, the 'higher rate') with a packet size at the physical layer of 704 bits, a 4.75 kb/s rate ($rate_B$, the ‘lower rate’) with packets of 560 bits, and an additional constant rate ($rate_C$, the ‘average rate’) corresponding to the average packet size of a perfectly adaptive transmission that uses both $rate_A$ and $rate_B$ according to the equation:

$$rate_C = \pi_B rate_B + \pi_G rate_A. \tag{7}$$

The same figure also illustrates the analytical performance of the proposed adaptive transmission scheme (the ‘adaptive’ curve), which uses $rate_A$ when the channel is good, and $rate_B$ when the channel is bad. Since packet errors occur almost only in the bad state the packet error rate of the adaptive solution is quite close to the one of the lower rate. The adaptive scheme, however, can exploit the good channel states by increasing the speech coding rate and thus the overall perceived quality. When the average curve is close to the higher curve, the adaptive scheme is coding speech at an average rate close to the higher rate.

4. SIMULATIONS AND RESULTS

Adaptive transmission of speech using the GSM Adaptive Multi-Rate (AMR) coder was tested over an IEEE 802.11 wireless channel using the NS-2 [18] network simulator. Several network conditions with Voice over IP connections, with and without interfering TCP traffic, were simulated under different channel error rates. The network simulator was modified to include a channel error model for the wireless link based on the Gilbert-Elliot two state Markov model [9].

At the application level speech is encoded with a different GSM AMR codec mode according to the istantaneous error rate of the wireless medium. The possible states associated to the channel are two: a “good” state with low error rate and a “bad” state with a higher probability of bit errors. For the former, the 12.2 kb/s speech coding rate is used, while for the latter the encoder produces an output rate of 4.75 kb/s. Sensing of the channel is performed for each speech frame; we assume perfect knowledge of the channel state at the transmitter side.

The speech payload is sent over the network using the Real-time Transport Protocol (RTP) [15] as defined in the recent RFC 3267 [16] that specifies the payload format to be used for AMR encoded speech signals. We use a bandwidth-efficient AMR payload for a single channel session carrying a single speech frame block: compressed speech bits are arranged in descending sensitivity order, ten control bits are present at the beginning of the payload to carry information about mode request (4 bits), last frame of the payload (1 bit), coding mode (4 bits), and damaged speech frame (1 bit); finally additional bits are added to the end as padding to make the payload byte aligned.

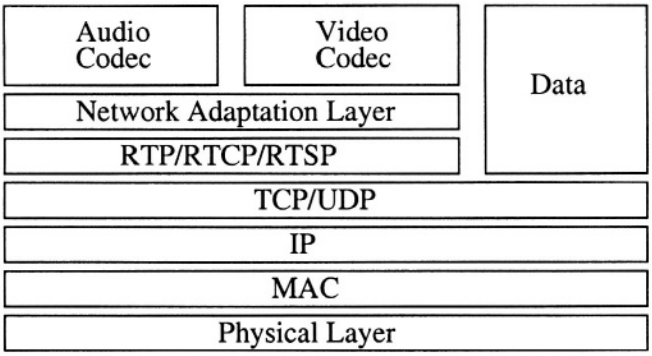


Figure 15-3. Protocol stack for IP-based real-time multimedia transmission over wireless LAN's.

Figure 15-3 presents the protocol stack used for the wireless real-time multimedia transmission. Every 20 ms the GSM AMR encoder produces a compressed speech payload of 14 bytes for the 4.75 kb/s mode and of 32 bytes for the 12.2 kb/s mode. The payload is then encapsulated by the RTP protocol, UDP is used for multiplexing different flows, and IP takes care of addressing and delivering the packets to their destination. For the RTP/UDP/IP headers, a compression scheme has been assumed that allows the 40-byte header to be compressed in 4 bytes as defined by RFC 3095 for Robust Header Compression [17]. Each data-type MPDU (MAC Protocol Data Unit) has a 24-byte header plus a 4-byte frame check sequence. Finally the physical level adds additional 24 bytes for the PLCP preamble, PLCP header, tail and pad bits.

4.1 Results

Simulations were performed for an 11 Mb/s IEEE 802.11 wireless LAN scenario where an infrastructure network is populated by three mobile terminals placed at the same distance from an access point (AP). The AP is then connected to another host through a wired link. All the communications are directed from the mobile stations to the wired node, and they are forwarded by the access point. Because the wireless path represents only the first transmission hop, we consider 20 ms as the maximum acceptable value for the one-way transfer delay over the wireless link: the percentage of voice packets that are lost or received with a delay greater than 20 ms is monitored on the access point.

Firstly, we tested the proposed adaptive solution against plain transmission of a single VoIP source without interfering traffic. A wireless node sends a speech

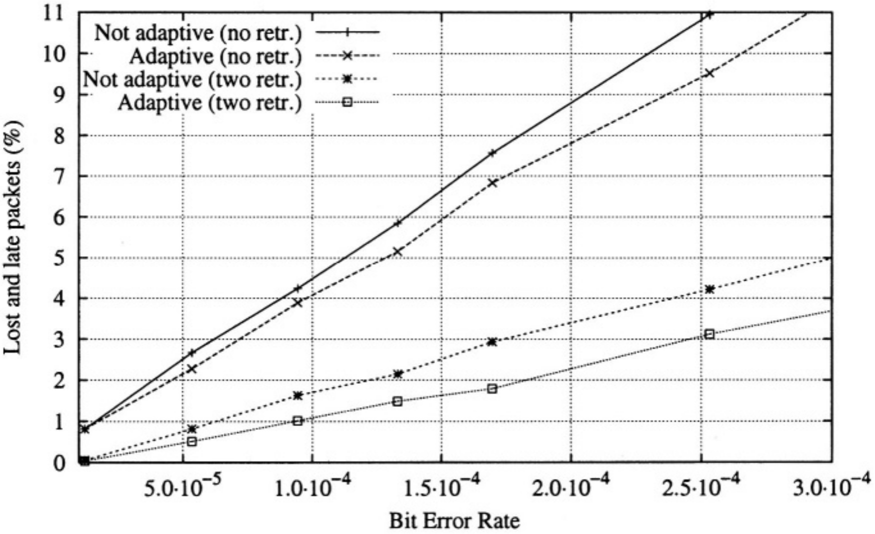


Figure 15-4. Lost and late speech packets as a function of the bit error rate; adaptive vs. not-adaptive technique, maximum number of retransmissions set to zero and two.

frame every 20 ms adapting the payload size (244 or 95 bits of speech data, which correspond to 704 or 560 bits at the physical layer) to the channel state. A bit error rate of 1×10^{-5} for the channel in the "good" condition, and a BER of 1.5×10^{-3} for the "bad" condition, that represents a channel fade, are used.

The transition probability p_{BG} of the Gilbert model has been kept constant at the value 0.6 so that the average sojourn time in the bad state (with a time slot of 20 ms) is $1/p_{BG} = 1.5$ time slots. The p_{BG} value has been changed to reflect different BER's as in Eq. 2.

Figure 15-4 compares adaptive and fixed-rate transmission at the same average bit-rate for the cases of maximum number of retransmissions zero and two, respectively. With a bit error rate at the physical level of 1.69×10^{-4} and two retransmissions, the adaptive solution nearly halves the number of lost and late packets from 2.9% to 1.7%. The plot also demonstrates that if the network allows a higher number of retransmissions the gap between constant bit rate transmission and adaptive transmission increases.

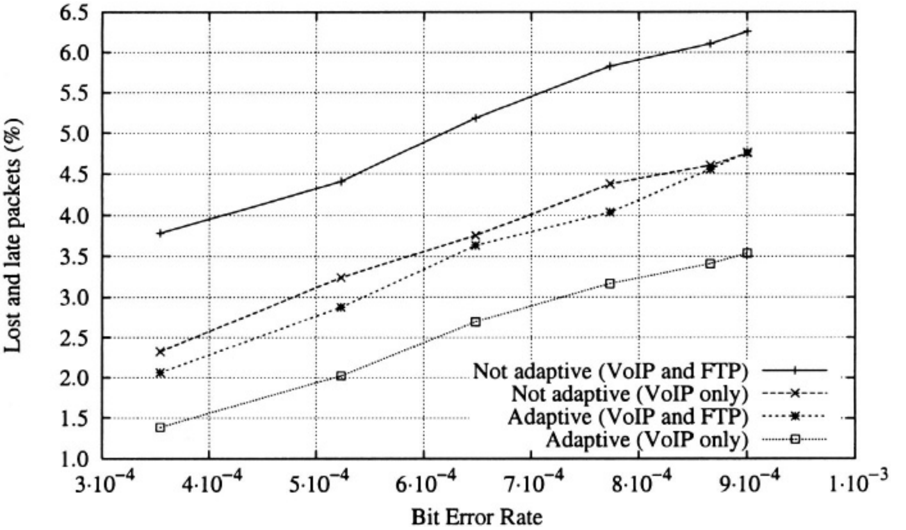


Figure 15-5. Lost and late speech packets as a function of the bit error rate; adaptive vs. not-adaptive technique, with and without interfering FTP traffic.

The second simulation scenario tests the performance of the adaptive transmission in presence of interfering FTP traffic. Besides the VoIP transmission two other terminals are sending FTP traffic to the wired host through the AP. The FTP packet size was set equal to the higher rate of the voice communication and its TCP congestion window was increased to make the source more aggressive in terms of used bandwidth against the VoIP connection. Figure 15-5 illustrates the case of two concurrent FTP sources with a maximum number of 4 retransmissions at the MAC level: the adaptive technique with interfering traffic performs better then the non-adaptive solution without interfering traffic. This is an important result because it shows that, in presence of network congestions, reducing the packet size by diminishing the source bit-rate is quite effective in improving the quality of interactive speech communications: a reduced source bit-rate decreases the congestion caused by packet retransmissions.

Regarding end-to-end delay, the proposed adaptive transmission scheme reduces the average delay because less retransmissions are needed. Figure 15-6

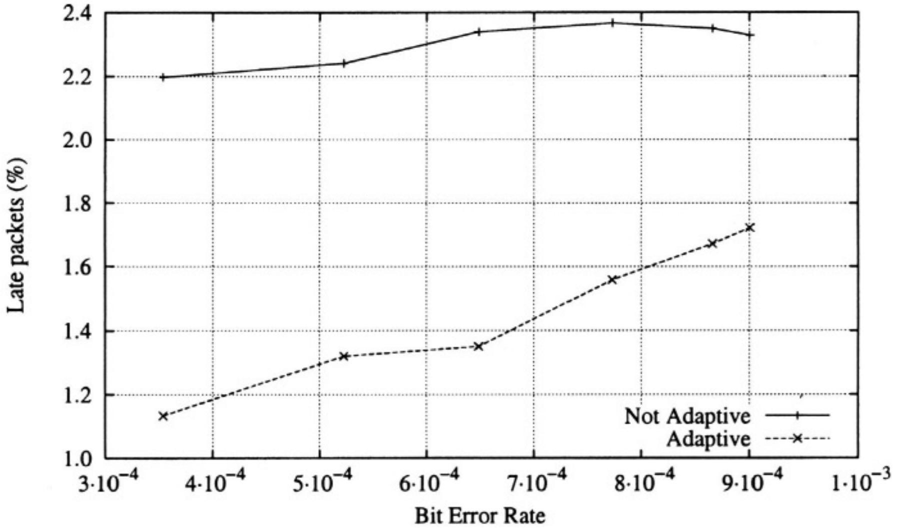


Figure 15-6. Packet discarded at the receiver because of their late arrival (delay larger than 20 ms) as a function of the bit error rate. Adaptive vs. not-adaptive technique.

shows the percentage of packets discarded at the receiver due to late arrival (delay > 20 ms). If the speech frame dimension is adapted to the channel condition, packets tend to arrive on time for successful playback, leading to higher perceptual quality.

Even more positive results were obtained for wideband speech transmission, for which a larger range of source rates is available. The GSM AMR wide-band codec [2] at rates of 23.85 kb/s and 6.6 kb/s was used in the same wireless scenario. The behavior of a single VoIP communication without interfering traffic is plotted in Figure 15-7 where the even larger advantage of the adaptive solution over the constant bit-rate transmission is clearly noticeable with respect to Figure 15-4: with BER equal to 1.69×10^{-4}, the packet loss rate is more than halved (from 4.3% to 2%).

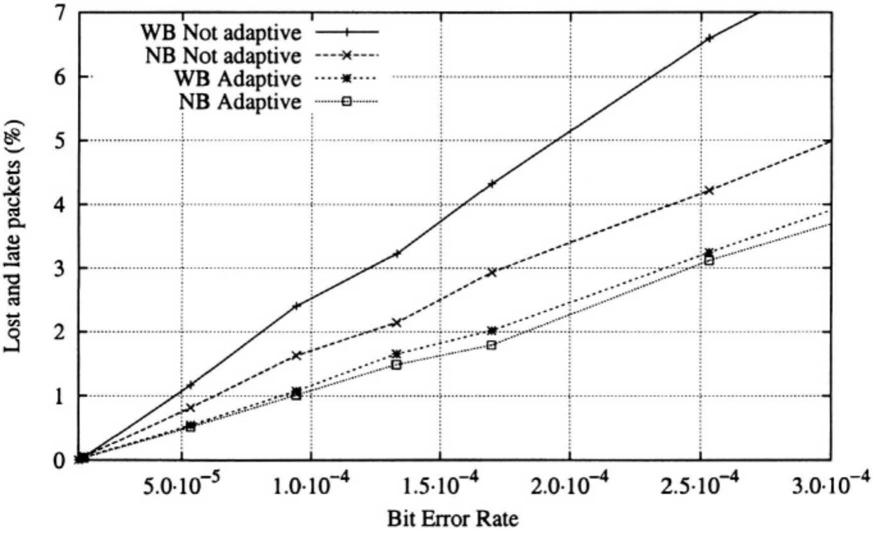


Figure 15-7. Lost and late speech packets as a function of the bit error rate; adaptive vs. not adaptive technique for narrowband and wideband speech, maximum number of retransmissions set to two.

5. CONCLUSIONS

An adaptive technique to transmit speech over 802.11 wireless packet networks was presented. The speech coding rate of a network-driven variable bit-rate coder, the GSM AMR, is selected to match the instantaneous wireless channel conditions: higher rates (i.e., larger packets) for low error rates, lower rates (i.e., smaller packets) when the channel is noisy. Network simulations showed that adaptively selecting the speech packet size consistently outperforms the constant bit-rate approach in terms of packet loss rates and end-to-end delays.

References

[1] ISO/IEC, 8802-11 (1999). Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11.*

[2] Bessette, B., Salami, R., Lefebvre, R., Jelinek, M., Rotola-Pukkila, J., Vainio, J., Mikkola, H., and Jarvinen, K. (2002). The adaptive multirate wideband speech codec (AMR-WB). *IEEE Transactions on Acoustics, Speech and Signal Processing,* 10(8):620–636.

[3] Blackard, K.L., Rappaport, T.S., and Bostian, C.W. (1993). Measurements and models of radio frequency impulsive noise for indoor wireless communications. *IEEE Journal on Selected Areas in Communications,* 11(7):991–1001.

[4] Bruhn, S., Blocher, P., Hellwig, K., and Sjoberg, J. (1999). Concepts and solutions for link adaptation and inband signaling for the GSM AMR speech coding standard. In *Proc. IEEE 49th Vehicular Technology Conference,* volume 3, pages 2451–2455.

[5] Christianson, L. and Brown, K. (1999). Rate adaptation for improved audio quality in wireless networks. In *Proc. IEEE Int. Workshop on Mobile Multimedia Communications,* pages 363–367, San Diego, CA , USA.

[6] Ci, S. and Sharif, H. (2002). An link adaptation scheme for improving throughput in the IEEE 802.11 wireless LAN. In *Proc. 27th Annual IEEE Conf. on Local Computer Networks,* pages 205–208, Flint, MI, USA.

[7] Crow, B.P., Widjaja, I., Kim, J.G., and Sakai, P.T. (1997). IEEE 802.11 Wireless Local Area Netoworks. *IEEE Communications Magazine,* 35(9):116–126.

[8] Ebert, J.P. and Willig, A. (1999). A Gilbert-Elliot bit error model and the efficient use in packet level simulation. *TKN Technical Reports Series.*

[9] Elliot, E.O. (1963). Estimates on error rates for codes on burst-noise channels. *Bell Syst. Tech. J.*, 42:1977–1997.

[10] ETSI (2000). Digital cellular telecommunications system (phase 2+); adaptive multi-rate (AMR) speech transcoding. *EN 301 704.*

[11] ETSI (2001). Digital cellular telecommunications system (phase 2+); link adaptation. *TS 101 709.*

[12] Homayounfar, K. (2003). Rate adaptive speech coding for universal multimedia access. *IEEE Signal Processing Magazine,* 20(2):30–39.

[13] Kaindl, M. and Gortz, N. (2002). AMR voice transmission over mobile internet. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing,* volume 2, pages 2049–2052, Orlando, Florida.

[14] Qiao, D., Choi, S., and Shin, K.S (2002). Goodput analysis and link adaptation for IEEE 802.11a wireless LAN's. *IEEE Transactions on Mobile Computing,* 1(4):278–292.

[15] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. (1996). RTP: A transport protocol for real-time applications. *RFC 1889.*

[16] Sjoberg, J., Westerlund, M., Lakaniemi, A., and Xie, Q. (2002). Real-time transport protocol (RTP) payload fromat and file storage format for the adaptive multi-rate (AMR) and adaptive multi-rate wideband (AMR-WB) audio codecs. *RFC 3267.*

[17] Svanbro, K. (2002). Lower layer guidelines for robust RTP/UDP/IP header compression. *RFC 3409.*

[18] UCB/LBNL/VINT (1997). Network Simulator – ns – version 2. *URL: http://www.isi.edu/nsnam/ns.*

[19] Veeraraghavan, M., Cocker, N., and Moors, T. (2001). Support of voice services in IEEE 802.11 wireless LAN's. In *Proceedings of INFOCOM,* pages 488–496.

[20] Wang, H.S. and Moayeri, N. (1995). Finite state markov channel - a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology,* 44(1): 163–171.

[21] Zorzi, M., Rao, R.R., and Milstein, L.B. (1998). Error statistics in data transmission over fading channels. *IEEE Transactions on Communications,* 46(11): 1468–1477.

This page intentionally left blank

Chapter 16

JOINT AUDIO-VIDEO PROCESSING FOR ROBUST BIOMETRIC SPEAKER IDENTIFICATION IN CAR[1]

Engin Erzin, Yücel Yemez, A. Murat Tekalp

Multimedia, Vision and Graphics Laboratory, College of Engineering, Koç University, Sarıyer, Istanbul, 34450, TURKEY; Email: eerzin@ku.edu.tr

Abstract: In this chapter, we present our recent results on the multilevel Bayesian decision fusion scheme for multimodal audio-visual speaker identification problem. The objective is to improve the recognition performance over conventional decision fusion schemes. The proposed system decomposes the information existing in a video stream into three components: speech, lip trace and face texture. Lip trace features are extracted based on 2D-DCT transform of the successive active lip frames. The mel-frequency cepstral coefficients (MFCC) of the corresponding speech signal are extracted in parallel to the lip features. The resulting two parallel and synchronous feature vectors are used to train and test a two stream Hidden Markov Model (HMM) based identification system. Face texture images are treated separately in eigenface domain and integrated to the system through decision-fusion. Reliability based ordering in multilevel decision fusion is observed to be significantly robust at all SNR levels.

Keywords: Speaker identification, multi-modal, multilevel decision fusion, robustness, in-vehicle

[1] This work has been supported by *The Scientific and Technical Research Council of Turkey* (TUBITAK) under the project EEEAG-101E038.

1. INTRODUCTION

Biometric person recognition technologies include recognition of faces, fingerprints, voice, signature strokes, iris and retina scans, and gait. Person recognition in general encompasses two different, but closely related tasks: Identification and verification. The former refers to identification of a person from her/his biometric data from a set of candidates, while the latter refers to verification of a person's biometric data. It is generally agreed that no single biometric technology will meet the needs of all potential recognition applications. Although the performances of these biometric technologies have been studied individually, there is relatively little work reported in the literature on the fusion of the results of various biometric technologies [1].

A particular problem in multi-modal biometric person identification, which has a wide variety of applications, is the speaker identification problem where basically two modalities exist: audio signal (voice) and video signal. Speaker identification, when performed over audio streams, is probably one of the most natural ways to perform person identification. However, video stream is also an important source of biometric information, in which we have still images of biometric features such as face and also the temporal motion information such as lip, which is correlated with the audio stream. Most speaker identification systems rely on audio-only data [2]. However especially under noisy conditions, such systems are far from being perfect for high security applications. The same observation is also valid for systems using only visual data; where poor picture quality or changes in lighting conditions significantly degrade performance [3,4]. A better alternative is the use of both modalities in a single identification scheme. Person identification has a variety of applications at various levels of security. A possible low security level application could be the identification of a specific driver/passenger in car that provides various personal control services to the driver or to the passenger. Speaker identification performance usually degrades under adverse environmental conditions such as car noise, and a multi-modal identification system helps to maintain a high level reliability for the driver/passenger identification task. The visual data could be available through a camera located on the visor.

The design of a multimodal identification system consists of two basic problems. The first problem is to represent the raw data acquired for each modality with a meaningful and robust set of features, which has to be individually able to discriminate samples belonging to different classes under varying environmental conditions. The second problem is to combine the decisions of individual classifiers so as to enforce the final decision. With the assumption that each selected feature set is individually discriminative enough under ideal conditions, the main motivation in a multimodal fusion scheme is to compensate possible misclassifications of a certain modality with other available modalities and to end up with a more reliable system. These misclassifications are in general inevitable due to environmental noise, measurement errors or time-varying characteristics of the signals. A critical issue in multimodal fusion is not to deteriorate the performance of unimodal classifiers. Thus *our ultimate goal should be at least not to fail whenever one of the individual classifiers gives the correct decision.* In this work, rather than selecting the best feature set, the emphasis is on this second problem, i.e. how to combine the decisions of different classifiers in view of the above discussion. We claim that the crucial point here is first to assess the reliability of each classifier, or modality, and then favor the classifiers according to their reliabilities in an appropriate decision fusion scheme.

Existing multimodal speaker identification systems are mostly bimodal, integrating audio and face information as in [8, 9, 10, 18], audio and lip information as in [11, 12, 13, 16, 19] or face and lip shape as in [14]. In [10], Sanderson et. al. present an audio-visual person verification system that integrates voice and face modalities and compares concatenative data-fusion with adaptive and non-adaptive decision fusion techniques, where adaptation takes into account the acoustic noise level of speech signal. Later in [8], enhanced PCA for face representation and fusion using SVMs and confidence measures are presented. Another audio-visual person identification system proposed in [9] uses a Maximum Likelihood Linear Transformation (MLLT) based data-fusion technique. These related works do not address lip-motion as a biometric modality for person identification and they all do emphasize on the performance of data and decision fusion in separate. In an eigenface-based person identification system, Kittler et. al use the lip-shape to classify face images to enhance the face recognition performance [14].

The only work in the literature that addresses a multimodal speaker identification system using speech, face and lip motion is the one presented in [19]. In [19], the information coming from voice, lip-motion and face modalities are assumed to be independent of each other and thus the multimodal classification is achieved by a decision fusion mechanism. The face-only module involves a quite deal of image analysis to normalize and to extract salient features of the face whereas the lip movement is represented by DCT coefficients of the corresponding optical flow vectors in the lip region. Face and lip features are then stored as biometric templates and classified through a set of algorithms so-called synergetic computer. The acoustic information on the other hand is represented by cepstral coefficients that are then classified by vector quantization using a minimum distance classifier.

In our biometric speaker identification system, we use three different modalities: speech, lip trace and face texture. Lip movement is a natural by-product of the speaking act. Information inherent in lip movement has so far been exploited mostly for the speech recognition problem, establishing a one-to-one correspondence with the phonemes of speech and the visemes of lip movement. It is quite natural to assume that lip movement would also characterize an individual as well as what that individual is speaking. Only few articles in the literature incorporate lip information for the speaker identification problem [11, 16, 19]. Although these works demonstrate some improvement over unimodal techniques, they use a decision-fusion strategy and hence do not fully exploit the mutual dependency between lip movement and speech. In a recent work [12], bimodal data and decision fusion of audio and eigenlip stream has been studied with encouraging results. In this chapter we present an HMM-based speaker identification scheme for joint use of the face, the lip trace and the audio signal of a speaking individual through data and multilevel decision fusion.

2. MULTIMODAL DECISION FUSION

The speaker identification problem is often formalized by using probabilistic approach: Given a feature vector $\boldsymbol{f}$ representing the sample data of an unknown individual, compute the a posteriori probability $\mathrm{P}(\lambda_n \mid \boldsymbol{f})$ for each class λ_n, $n = 0,1,\ldots,N,$ i.e. for each speaker's model. The sample feature vector is then assigned to the class $\lambda *$ that maximizes the a posteriori probability:

$$\lambda * = \arg\max_{\lambda_n} \mathrm{P}(\lambda_n \mid \boldsymbol{f}). \tag{1}$$

Since $P(\lambda_n \mid f)$ is usually difficult to compute, one can rewrite (1) in terms of class-conditional probabilities. Using Bayes Rule, we have

$$P(\lambda_n \mid f) = \frac{P(f \mid \lambda_n)P(\lambda_n)}{P(f)}. \tag{2}$$

Since P(*f*) is class independent and assuming equally likely class distribution, $P(\lambda_n) = \frac{1}{N}$, Eq. (1) is equivalent to

$$\lambda^* = \arg\max_{\lambda_n} P(f \mid \lambda_n). \tag{3}$$

Computation of class-conditional probabilities $P(f \mid \lambda_n)$ needs a prior modeling step, through which a probability density function of feature vectors is estimated for each class by using available training data. This modeling step is also referred to as training phase.

In a speaker identification scheme, a reject mechanism is also required due to possible impostor identity claims. A possible reject strategy is thus to refer a reject (imposter) class $\lambda_{\bar{n}}$, so that a likelihood ratio $\rho(f \mid \lambda_n)$ in logarithmic domain is used for accept or reject decision:

$$\rho(f \mid \lambda_n) = \log\frac{P(f \mid \lambda_n)}{P(f \mid \lambda_{\bar{n}})} = \log P(f \mid \lambda_n) - \log P(f \mid \lambda_{\bar{n}}). \tag{4}$$

Ideally, the imposter class model should be constructed by using all possible imposter observations for class *n*, which is practically unfeasible to achieve. In this work we use the universal background model, which is estimated, by using all available training data regardless of which class they belong to. The final decision strategy can be stated as follows:

$$\begin{array}{ll} \text{if } \rho(f \mid \lambda^*) \geq \tau & \text{accept;} \\ \text{otherwise} & \text{reject,} \end{array} \tag{5}$$

where τ is the optimal threshold which is usually determined experimentally to achieve the desired false accept or false reject rate.

When two or more modalities exist, the selection of the appropriate fusion technique, whether data or decision fusion, should take into account how these modalities are correlated to each other. In the case of decision fusion, i.e. when the modalities are uncorrelated, a critical issue is that individual class-conditional probabilities, and the log-likelihood ratios as well, usually results in values with different ranges, with different means and variances. Thus prior to the fusion process, a common practice is to apply normalization on resulting likelihoods, such as sigmoid normalization. Another issue is varying reliability of each likelihood contributing to the final decision. Thus commonly, a weighted sum of normalized likelihoods is used:

$$\rho(f_1,\ldots,f_P \mid \lambda_n) = \sum_{p=1}^{P} \omega_p \rho(f_p \mid \lambda_n), \tag{6}$$

where ω_p values are weighting coefficients to be determined. Most of the decision fusion schemes existing in the literature [15, 17] vary actually in the way they interpret Equation (6). In one extreme, there are techniques that try to estimate these coefficients, which are ideally feature and class dependent. Coefficients can be set to some fixed values using some a priori knowledge or can be estimated adaptively via various methods such as noise estimation or measuring the experimental or statistical dispersion of each decision [11]. The problem of this approach is that, estimation of the reliability parameters itself is not in general very reliable and moreover unimodal misclassification may occur even with high likelihood ratios. Erroneous decisions keep contributing to the final decision likelihood, hence scarifying from correct unimodal decisions. In the other extreme, there are techniques based on the well-known *max* rule [15]. In regard to Equation (6), this strategy uses the following rule to set the coefficients ω_p:

$$\omega_p = \begin{cases} 1 & \text{if} \quad p = \arg\max_i \rho(f_i \mid \lambda_n) \\ 0 & \qquad\qquad \text{otherwise} \end{cases} \tag{7}$$

When seen as a binary mechanism as above, the max rule may filter out most of the erroneous contributions in the final decision. But the fact that unimodal misclassification may occur even with high likelihood ratios is still not taken into account. In the next subsection, we propose a decision scheme that compromises these two extreme approaches.

2.1 Multilevel Bayesian Decision Fusion

Looking back to Equation (5), once a threshold τ is set in the likelihood ratio test, one can claim that if the log likelihood ratio $\rho(f \mid \lambda_n)$ is much larger or much smaller than τ, the confidence of the decision is stronger. Hence the absolute difference between the likelihood ratio $\rho(f \mid \lambda_n)$ and the threshold τ can be used as a measure of confidence C_f,

$$C_f = \left|\rho(f \mid \lambda_n) - \tau\right|. \qquad (8)$$

In the multimodal scenario, the confidence measure can be used beneficially in the decision fusion if we have enough a priori information on the different modality streams. Let us define a multimodal scenario with three different modalities and feature vectors f_1, f_2 and f_3. There are also three streams of log likelihood ratios $\rho(f_1 \mid \lambda^*)$, $\rho(f_2 \mid \lambda^*)$ and $\rho(f_3 \mid \lambda^*)$, correspondingly. If we have a priori information such that the reliability of the modalities are in an order, such that the first modality f_1 is the most reliable and the last modality f_3 is the least reliable source under some controlled conditions (such as low acoustic noise, frontal face stream, etc.), then the confidence of the decision that is coming from first modality as defined in (8) would be beneficial for the decision fusion. Keeping this fact in mind a Bayesian decision system can be built. In this system a decision tree is utilized as:

a) A decision (accept or reject) is taken according to the f_1 modality if the confidence measure C_{f_1}, that is coming from the most reliable modality f_1, is high enough (i.e. if $C_{f_1} > \tau_1$),
b) Otherwise a decision is taken according to the modality with the highest confidence among f_1 and f_2 if $C_{f_2} > \tau_2$,
c) Otherwise a decision is taken according to the modality with the highest confidence among all three modalities.
d) Note that the decision scheme uses three confidence thresholds τ, τ_1 and τ_2 that have to be determined experimentally.

2.2 WTAll Decision Fusion

The conventional max rule given by Equation (7) can be modified so as to better handle possible false identity claims. In this slightly modified scheme that we will refer to as winner modality takes all (WTAll), the likelihood ratios in (7) are substituted with confidence measures as defined in (8):

$$\omega_p = \begin{cases} 1 & \text{if} \quad p = \arg\max_i \left| \rho(f_i \mid \lambda_n) - \tau \right| \\ 0 & \text{otherwise} \end{cases}$$

In this manner, a strong decision for rejection can also be taken into account and favored even though the corresponding likelihood ratio is not the maximum of the likelihoods resulting from all available modalities.

3. FEATURE EXTRACTION

In this section we consider a text-dependent multimodal speaker identification. The bimodal database consists of audio and video signals belonging to individuals of a certain population. Each person in this database utters a predefined secret phrase that may vary from one person to another. The objective is, given the data of an unknown person, to find whether this person matches someone in the database or not. The person is identified if there is a match and is rejected if not. The multimodal system uses three feature sets extracted from each audio-visual stream that correspond to three modalities: Face, lip trace and speech. Our goal is at least not to fail whenever one of the individual classifiers gives the correct decision and also to be robust against false identity claims. The overall classification is based on the theoretical framework presented in Section 2.

3.1 Face Modality

The eigenface technique [4], or more generally the principal component analysis, has proven itself as an effective and powerful tool for recognition of still faces. The core idea is to reduce the dimensionality of the problem by obtaining a smaller set of features than the original dataset of intensities. In principal component analysis, every image is expressed as a linear combination of some basis vectors, i.e. eigenimages that best describe the variation of intensities from their mean. When a given image is projected onto this lower dimensional eigenspace, a set of r eigenface coefficients is obtained, that gives a parameterization for the distribution of the signal. Obtaining principal components of an image signal, i.e. eigenimages, can be thought of as an eigenvalue problem. Suppose that the training set consists of M mean-removed image vectors $\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_M$. Then the eigenimages $\boldsymbol{v}_m$, m = 0, 1,..., M, can be computed as the eigenvectors of the following covariance matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{x}_m \boldsymbol{x}_m^{\mathrm{T}} .$$

Each eigenimage $\boldsymbol{v}_m$ is associated to an eigenvalue and principal components are given by the first R eigenimages associated to the first R eigenvalues when ordered with respect to their magnitudes. Usually the reduced dimension R is much smaller than M, and the r-th eigenimage coefficient w_r, is obtained by the projection $w_r = \boldsymbol{v}_r^{\mathrm{T}} \boldsymbol{y}$ for a given test image vector $\boldsymbol{y}$.

The eigenface coefficients, when computed for every frame i of a given test sequence, constitute the face texture feature vector $\boldsymbol{f}_{\mathrm{F}}^{i} = [w_1, w_2, \ldots, w_R]$, $i = 1, \ldots, K$. The face images in the training set are all used first to obtain the eigenspace. The training set contains a number of image sequences, say L, from each speaker class λ_n. Let $\boldsymbol{f}_{\mathrm{F}n}^{j}$, $j = 1, \ldots, K \cdot L$, denote the feature vectors of these images belonging to the class λ_n in the training set. Then the minimum distance d_n between these two sets of feature vectors can be used as a similarity metric between the speaker class λ_n and the unknown person:

$$d_n = \min_{i,j} \left\| \boldsymbol{f}_{\mathrm{F}}^{i} - \boldsymbol{f}_{\mathrm{F}n}^{j} \right\|. \tag{9}$$

The similarity metric defined in (9) can also be expressed as a probabilistic likelihood by making use of Gibbs distribution: Given the face texture feature vectors $\boldsymbol{f}_{\mathrm{F}}^{1}, \boldsymbol{f}_{\mathrm{F}}^{2}, ..., \boldsymbol{f}_{\mathrm{F}}^{K}$, the class conditional probability of the feature set can be written as

$$\mathrm{P}(f_{\mathrm{F}}^{1}, f_{\mathrm{F}}^{2}, \ldots, f_{\mathrm{F}}^{K} \mid \lambda_{n}) = \frac{1}{\kappa} \mathrm{e}^{-d_{n}/\sigma} \quad , \tag{10}$$

where $\kappa = \sum_{d} \mathrm{e}^{-d/\sigma}$ and σ is the decay coefficient of the Gibbs distribution function, that can be used for likelihood normalization. The log likelihood ratio is then defined as:

$$\begin{aligned} \rho(f_{\mathrm{F}}^{1}, f_{\mathrm{F}}^{2}, \ldots, f_{\mathrm{F}}^{K} \mid \lambda_{n}) &= \log \mathrm{P}(f_{\mathrm{F}}^{1}, f_{\mathrm{F}}^{2}, \ldots, f_{\mathrm{F}}^{K} \mid \lambda_{n}) - \log \mathrm{P}(f_{\mathrm{F}}^{1}, f_{\mathrm{F}}^{2}, \ldots, f_{\mathrm{F}}^{K} \mid \lambda_{\bar{n}}) \\ &= \frac{\tilde{d} - d_{n}}{\sigma} \end{aligned} \tag{11}$$

The log likelihood ratio as defined in Equation (11) requires the definition of a universal background class $\lambda_{\bar{n}}$. For this, we will adapt the faceness measure defined by the authors in [4]. The eigenspace origin will be used as the representative feature vector of the face universal background class. Hence $\tilde{d}$ is defined as the distance of the feature vector f_{F}^{i} (that yields the minimum distance d_{n}) to the universal background model. The log likelihood ratio in (11) is computed for each class λ_{n}, and can be fused with decisions coming from other available modalities.

3.2 Audio and Lip Modalities

The two synchronized modality streams, audio and lip, are used separately and jointly to extract reliable identification performance under varying environmental conditions. Audio features and lip features are extracted separately from these synchronized streams at different rates. Hence a rate adjustment is needed when these two modalities are jointly fused to each other.

The audio stream is represented with the mel-frequency cepstral coefficients (MFCC), as they yield good discrimination of speech signal. In our system, the speech signal sampled at 16 kHz is analyzed on 25 ms frame basis by frame shifts of 10 ms. Each frame is first multiplied with a Hamming window and transformed to frequency domain using Fast Fourier Transform (FFT). Mel-scaled triangular filter-bank energies are calculated over the square magnitude of the spectrum and represented in logarithmic scale [5]. The resulting MFCC features are derived using discrete cosine transform over log-scaled filter-bank energies e_i:

$$c_j = \frac{1}{N_M} \sum_{i=1}^{N_M} e_i \cos((i-0.5)\frac{j\pi}{N_M}) \quad \text{for } j = 1,2,\ldots,N \tag{12}$$

where N_M is the number of mel-scaled filter banks and N is the number of MFCC features that are extracted. The MFCC feature vector for the k-th frame is defined as, $\boldsymbol{C}_k = [c_1 c_2 \cdots c_N]^{\mathrm{T}}$. The audio feature vector f_{A}^k for the k-th frame is formed as a collection of MFCC vector $\boldsymbol{C}_k$ along with the first and second delta MFCCs, $f_{\mathrm{A}}^k = [\boldsymbol{C}_k \;\; \Delta\boldsymbol{C}_k \;\; \Delta\Delta\boldsymbol{C}_k]$.

The gray scale intensity based lip stream is transformed into 2D-DCT domain and then each lip frame is represented by the first M DCT coefficients of the zig-zag scan excluding the 0-th dc coefficient. The lip feature vector for the i-th lip frame is denoted by f_{L}^i. As the audio features are extracted at a rate of 100 fps and the lip features are extracted at a rate of 15 fps, rate synchronization should be performed prior to the data fusion. The lip features are computed using linear interpolation over the f_{L}^i sequence to match the 100 fps rate as follows:

$$\tilde{f}_{\mathrm{L}}^k = (1-\alpha_k) f_{\mathrm{L}}^{i^*} + \alpha_k f_{\mathrm{L}}^{i^*+1}$$

where $i^* = \left\lfloor \frac{3k}{20} \right\rfloor$ and $\alpha_k = \frac{3k}{20} - i^*$.

The unimodal and fused temporal characterizations of the audio and the lip modalities are performed using Hidden Markov Models, which are reliable structures to model human hearing system, and thus they are widely used for speech recognition and speaker identification problems [2]. In this work a word-level continuous-density HMM structure is built for the speaker identification task. Each speaker in the database population is modeled using a separate HMM and is represented with the feature sequence that is extracted over the audio/lip stream while uttering the secret phrase. First a world HMM model is trained over the whole training data of the population. Then each HMM associated to a speaker is trained over some repetitions of the audio-video utterance of the corresponding speaker. In the identification process, given a test feature set, each HMM structure produces likelihood.

These likelihoods along with the likelihoods of the HMM representing the world class results in a log likelihood ratio to be used in the multimodal decision fusion.

Two possible audio-lip fusion schemes are carried out using concatenative data fusion [12] and multi-stream HMMs [20]. The concatenative data fusion is based on the early integration model [7] where the integration is performed in the feature space to form a composite feature vector of audio and lip features. Hence the joint audio-lip feature $\boldsymbol{f}_{\mathrm{AL}}^{k}$ is formed by combining the audio feature $\boldsymbol{f}_{\mathrm{A}}^{k}$ and the interpolated lip features $\tilde{\boldsymbol{f}}_{\mathrm{L}}^{k}$ for the k-th audio-visual frame: $\boldsymbol{f}_{\mathrm{AL}}^{k} = [\boldsymbol{f}_{\mathrm{A}}^{k} \quad \tilde{\boldsymbol{f}}_{\mathrm{L}}^{k}]$.

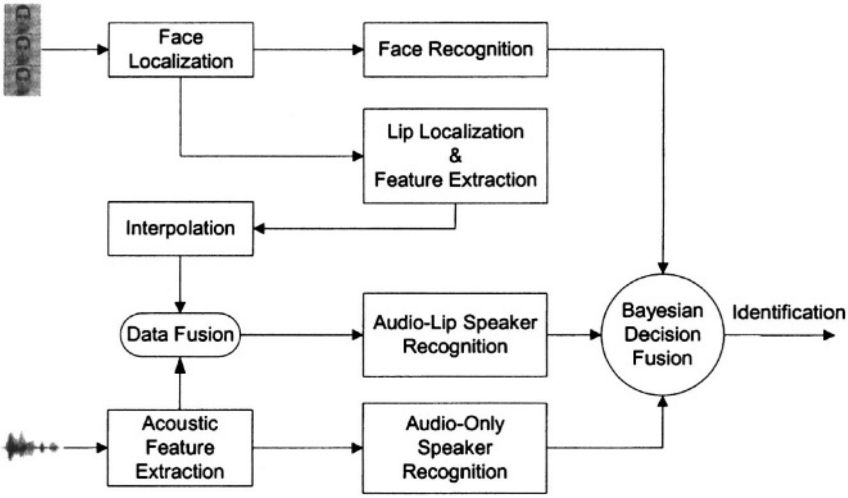

Face Localization
Face Recognition
Lip Localization & Feature Extraction
Interpolation
Data Fusion
Audio-Lip Speaker Recognition
Bayesian Decision Fusion
Identification
Acoustic Feature Extraction
Audio-Only Speaker Recognition

Figure 16-1. Multimodal speaker identification system.

4. EXPERIMENTAL RESULTS

The block diagram for the multimodal audio-visual speaker identification system is given in Figure 16-1. The database that has been used to test the performance of the proposed speaker identification system includes 50 subjects. Training and testing are performed over two independent set of recordings with each having five repetitions. A set of impostor data is also collected with each subject in the population uttering five different names from the population. The audio-visual data MVGL-AVD have been acquired using a Sony DSR-PD150P video camera at Multimedia Vision and Graphics Laboratory of Koç University. A collection of sample images from the audio-visual database is presented in Figure 16-2.

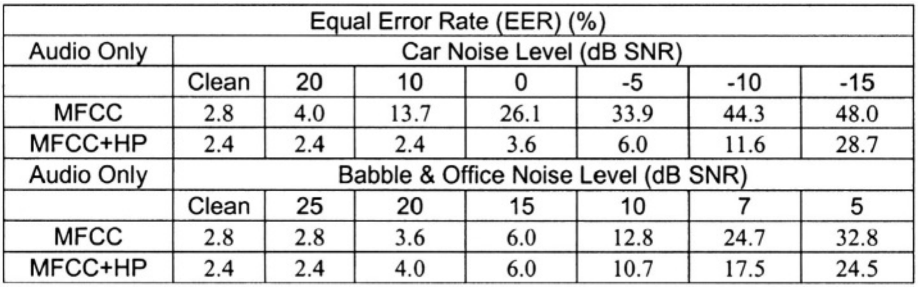
Equal error rate (EER), where false accept rate (FAR) equals false reject rate (FRR) operating point, is used in the performance analysis of speaker identification system. False accepts occur when an imposter is identified as an accepted client or when a client from the accept database identified incorrectly. False rejects occur when a client from the accept database is rejected. The false accept and the false reject rates are computed as,

$$\mathrm{FAR} = 100 \times \frac{\#\text{of false accepts}}{N_a + N_r} \text{ and } \mathrm{FRR} = 100 \times \frac{\#\text{of false rejects}}{N_a},$$

where N_a and N_r are the total number of trials in the accept and reject scenario, respectively.

The temporal characterization of audio, lip and audio-lip fused streams have been obtained using a 6-state left-to-right two-mixture continuous density HMM structure for each speaker. The acquired video data is first split into segments of secret phrase utterances. The visual and audio streams are then separated into two parallel streams, where the visual stream has gray-level video frames of size 720×576 pixels containing the frontal view of a speaker's head at a rate of 15 fps and the audio stream has 16 kHz sampling rate. The acoustic noise, which is added to the speech signal to observe the identification performance under adverse conditions, is picked to be either car noise or a mixture of office and babble noise.

The audio stream processing is done over 10 ms frames centered on 25 ms Hamming window. The MFCC feature vector, C_k, is formed from 13 cepstral coefficients excluding the 0^{th} gain coefficient using 26 mel-frequency bins. The resulting audio feature vector, f_A^k of size 39, includes the MFCC vector and the first and the second delta MFCC vectors. Two variations of the audio feature vector are defined based on the frequency selectiveness in MFCC calculation. The first mel-band that calculates the first energy term e_1 (see Eq. 12) is picked to start at 50 Hz and at 250 Hz, where these features are called MFCC and high-pass MFCC (MFCC+HP), respectively. In the audio-only scenario the identification performance degrades rapidly with decreasing SNR as seen from Table 16-1. The high-pass MFCC features are observed to be more robust under environmental noise. The performance of the speaker identification system significantly increases with MFCC+HP feature set, especially under car noise, as car noise is spectrally concentrated at low frequencies. We even observe some performance improvement under clean conditions with the high-pass MFCC features, as low frequency contributions do not convey significant information for speaker identification.



Equal Error Rate (EER) (%)							
Audio Only	Car Noise Level (dB SNR)						
	Clean	20	10	0	-5	-10	-15
MFCC	2.8	4.0	13.7	26.1	33.9	44.3	48.0
MFCC+HP	2.4	2.4	2.4	3.6	6.0	11.6	28.7
Audio Only	Babble & Office Noise Level (dB SNR)						
	Clean	25	20	15	10	7	5
MFCC	2.8	2.8	3.6	6.0	12.8	24.7	32.8
MFCC+HP	2.4	2.4	4.0	6.0	10.7	17.5	24.5

Table 16-1. Equal error rate performances of the audio only speaker identification system with the MFCC and MFCC+HP features

Each video stream is around 1 second in duration and during this time it is assumed that the subject does not considerably move her/his head. Hence, detected lip regions are used to crop 64×40 lip frames to form the lip sequence of each visual stream. The lip feature vectors f_L^i, which are used in both training and testing of the HMM-based classifier, are obtained as described in Section 3.2 with M = 60. As for the extraction of face feature vectors, an eigenspace of dimension R = 20 is computed using two pictures from each utterance in the training part of the face sequence set. A sample face image and corresponding lip sequence is presented in Figure 16-3.

A summary of the modalities together with the decision fusion techniques is given in Table 16-2. Face-only and lip-only equal error rates are found to be 8.40% and 20.0%, respectively. The lip-only performance has a decent equal error rate. For the face-only case, we have to point out that the images in the training and testing set have varying backgrounds and lightings; this is why the face-only identification performance may seem to be worse than expected.

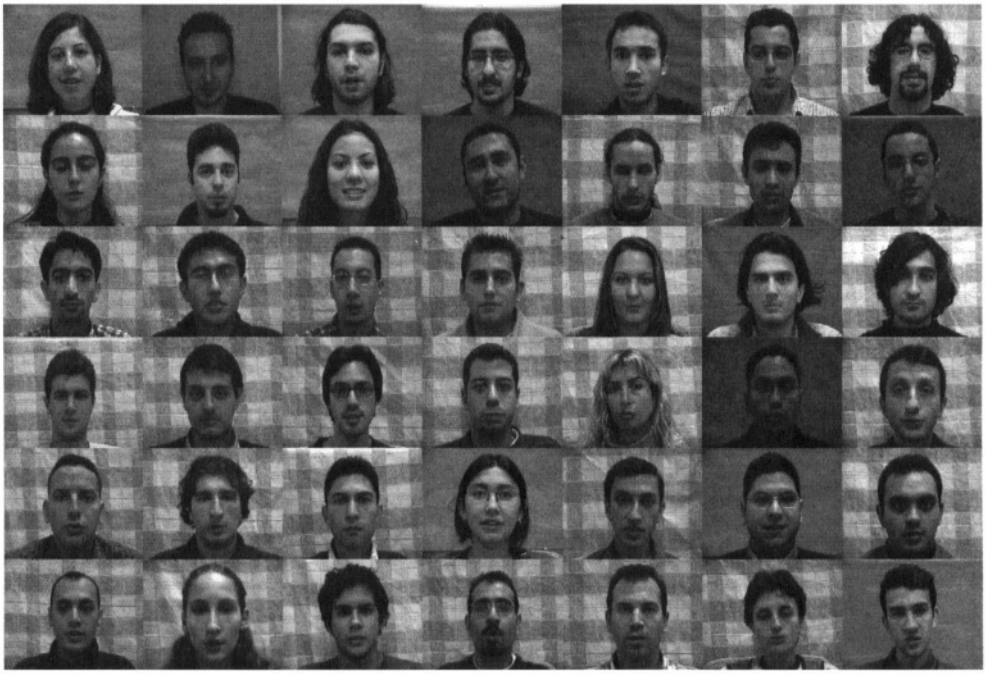


Figure 16-2. Selected sample images from the MVGL-AVD database.

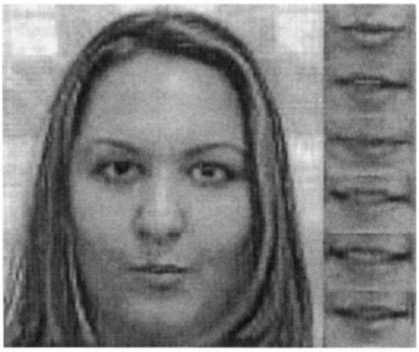


Figure 16-3. A sample image from the database and six frames from the corresponding lip sequence.

A	Audio only modality
L	Lip only modality
F	Face only modality
Aldf	Audio-Lip data fusion with concatenation
ALms	Audio-Lip with multi-stream HMMs (audio & lip weights are 0. &0.3, respectively)
+	Decision fusion with weighted sums ($\omega_k = 0.6\omega_{k-1}$ for k =2,…,P such that $\omega_1 + \cdots + \omega_P = 1$)
	Decision fusion with WTAll
•	Multilevel Bayesian decision fusion (leftmost being the most reliable modality)

Table 16-2. Abbreviations and descriptions for modalities and fusion techniques.

The multimodal identification results are shown in Tables 16-3 and 16-4, where we observe the equal error rates at varying levels of acoustic noise. Table 16-3 displays the equal error rates obtained for audio-lip fusion, which is based on concatenative data fusion and two-stream HMM structure. Although the performance figures for the audio-lip streams do not convey significant improvement and stay well under audio-only performances, they bring some independent information to the decision fusion using audio-lip correlations, especially under environmental noise. Decision fusion results are presented in Table 16-4, where summation, WTAll and Bayesian types of decision fusion techniques are evaluated.

Decision fusion techniques significantly improve identification rates, as they address the independence between different modalities. It is clear in Table 16-4 that the summation rule suffers under low SNR conditions. Although the weights in the summation rule are picked to be optimal with the existing modality reliabilities, the poor improvement under noise is mainly due to the variations of the reliabilities under adverse environmental conditions. However, WTAll decision fusion favors the confident modality, and performs better for low SNR conditions. On the other hand, multilevel Bayesian decision fusion favors confident enough modality if it stands higher in the reliability ordering, which introduces further improvement over all SNR conditions. In the multilevel Bayesian fusion, the reliability ordering of the modalities decreases from left to right. For example, in the Bayesian fusion A • F • ALms, the most reliable and the least reliable streams are the audio-only and the multi-stream audio-lip, respectively. The reliability orderings are assigned considering their single modality performances. The most promising decision fusion can be set as (A+F+ALms)•A•F, where weighted summation A+F+ALms is picked to have the most reliable source of information as it performs better under high SNR conditions, and audio-only and face-only are picked to be the other two modalities for multilevel Bayesian decision fusion. The benefit of multilevel Bayesian decision fusion is clear from the performances of A • F • ALms and (A+F+ALms)•A•F. Even though weighted summation achieves high performance results for multimodal systems, we observe further performance improvements using Bayesian decision tree over different likelihood streams with the prior knowledge of associated stream reliabilities.

Equal Error Rate (EER) (%)							
Modality	Car Noise Level (dB SNR)						
	Clean	20	10	0	-5	-	-
ALdf	18.5	18.5	18.6	18.8	19.0	19.2	19.6
ALms	16.0	15.9	16.0	16.0	16.2	16.5	17.6
	Babble & Office Noise Level (dB SNR)						
	Clean	25	20	15	10	7	5
ALdf	18.5	18.6	18.8	19.0	19.5	19.8	20.0
ALms	16.0	16.2	16.4	16.5	17.6	17.3	18.0

Table 16-3. Equal error rate performances of the audio-lip speaker identification systems at varying acoustic noise levels.

5. CONCLUSIONS

We have presented a multimodal (audio-lip-face) speaker identification system that improves the identification performance over unimodal schemes. These three independent sources of information with different reliabilities are put together to propose a reliability ordering based multilevel decision fusion. We observed significant improvement with WTAll decision fusion, and a further improvement is achieved using the multilevel Bayesian decision fusion. The reliability ordering is fixed with respect to the EER performances of individual likelihood sequences under acoustically clean conditions.

	Equal Error Rate (EER) (%)						
Decision Fusion	Car Noise Level (dB SNR)						
	Clean	20	10	0	-5	-10	-15
A + F + ALms	0.8	0.8	0.8	1.2	3.6	8.4	19.9
A F ALms	1.8	1.8	1.8	2.0	2.8	3.7	5.9
A • F	1.4	1.4	1.4	1.4	1.8	4.0	7.4
A • F • L	1.2	1.2	1.2	1.2	1.6	3.6	6.3
A • F • ALms	1.2	1.2	1.2	1.2	1.6	2.8	5.2
(A+F+ALms)•A• F	0.4	0.4	0.4	0.7	1.6	3.7	8.7
	Babble & Office Noise Level (dB SNR)						
	Clean	25	20	15	10	7	5
A + F + ALms	0.8	0.8	1.2	4.4	6.8	12.8	15.7
A F ALms	1.8	1.6	2.0	3.2	4.8	5.2	5.5
A • F	1.2	1.6	2.0	3.2	6.0	8.4	8.4
A • F • L	1.2	1.2	1.2	2.0	4.4	5.2	5.9
A • F • ALms	1.2	1.2	1.2	2.0	4.0	4.6	5.6
(A+F+ALms)•A• F	0.4	0.4	0.8	1.9	3.5	5.4	8.1

Table 16-4. Equal error rate performances of the speaker identification systems for various decision fusion techniques at varying acoustic noise levels.

6. CONCLUSIONS

We have presented a multimodal (audio-lip-face) speaker identification system that improves the identification performance over unimodal schemes. These three independent sources of information with different reliabilities are put together to propose a reliability ordering based multilevel decision fusion. We observed significant improvement with WTAll decision fusion, and a further improvement is achieved using the multilevel Bayesian decision fusion. The reliability ordering is fixed with respect to the EER performances of individual likelihood sequences under acoustically clean conditions. However we should note that this reliability ordering is not optimal under varying environmental conditions. Hence a better alternative is to adaptively predict the reliability of each modality, so that an optimal reliability ordering can be achieved for multilevel Bayesian decision fusion.

Robust estimation of reliabilities is yet an important and challenging problem, which is currently under investigation for future enhancements.

REFERENCES

[1] N.K. Ratha, A. Senior, and R.M. Bolle, Lecture Notes in Computer Science: Advances in Pattern Recognition - ICAPR 2001: Second International Conference Rio de Janeiro, Brazil, March 11-14, 2001, Proceedings, chapter Automated Biometrics, pp. 445-474, Springer-Verlag Heidelberg, January 2001.

[1] J.P. Campbell, "Speaker recognition: A tutorial," Proceedings of the IEEE, vol. 85, no. 9, pp. 1437-1462, September 1997.

[2] J. Zhang , Y. Yan and M. Lades, "Face recognition: Eigenface, elastic matching, and neural nets," Proceedings of the IEEE, vol. 85, no. 9, pp. 1423-1435, September 1997.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 586-591, September 1991.

[4] L. Rabiner and B-H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.

[5] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips, "Face recognition: A literature survey," UMD CfAR Technical Report, pp. CAR-TR-948, 2000.

[6] D. D. Zhang, Automated Biometrics, Kluwer Academic Publishers, 2000.

[7] C. Sanderson, S. Bengio, H. Bourlard, J. Mariethoz, R. Collobert, M.F. BenZeghiba, F. Cardinaux, and S. Marcel, "Speech and face based biometric authentication at IDIAP," Proc. of the Int. Conf. on Multimedia & Expo 2003 (ICME2003), vol. 3, pp. 1-4, July 2003.

[8] U. V. Chaudhari, G. N. Ramaswamy, G. Potamianos, and C. Neti, "Information fusion and decision cascading for audio-visual speaker recognition based on time-varying stream reliability prediction," Proc. of the Int. Conf. on Multimedia & Expo 2003 (ICME2003), vol. 3, pp. 9-12, July 2003.

[9] C. Sanderson and K. K. Paliwal, "Noise compensation in a person verification system using face and multiple speech features," Pattern Recognition, vol. 36, no. 2, pp. 293-302, February 2003.

[10] T. Wark and S.Sridharan, "Adaptive fusion of speech and lip information for robust speaker identification," Digital Signal Processing, vol. 11, no. 3, pp. 169-186, July 2001.

[11] A. Kanak, E. Erzin, Y. Yemez, and A. M. Tekalp, "Joint audio-video processing for biometric speaker identification," Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing 2003 (ICASSP 2003), vol. II, pp. 377-380, 2003.

[12] M. R. Civanlar and T. Chen, "Password-free network security through joint use of audio and video," Proceedings of SPIE Photonic, pp. 120-125, November 1996.

[13] J. Kittler, Y. P. Li, J. Matas, and M. U. Ramos Sanchez, "Lip-shape dependent face verification," First International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA), pp. 61-68, March 1997.

[14] J. Kittler, M. Hatef, R.P.W Duin, and J. Matas, "On Combining Classifiers", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226 - 239, 1998.

[15] P. Jourlin, J. Luettin, D. Genoud, and H. Wassner, "Acoustic-labial speaker verification," Pattern Recognition Letters, vol. 18, no. 9, pp. 853-858, 1997.

[16] H. Altincay and M. Demirekler, “An information theoretic framework for weight estimation in the combination of probabilistic classifiers for speaker identification,” Journal of Speech Communication, vol. 30, pp. 255-272, 2000.

[17] S. Ben-Yacoub, Y. Abdeljaoued and E. Mayoraz, “Fusion of face and speech data for person identity verification”, IEEE Trans. Neural Networks, vol. 10, no. 5, pp. 1064-1075, 1999.

[18] R.W. Frischholz and U. Dieckmann, “BioID: A multimodal biometric identification system,” Journal of IEEE Computer, vol. 33, no. 2, pp. 64-68, February 2000.

[19] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, and D. Vergyri, “Large-vocabulary audio-visual speech recognition: A summary of the Johns Hopkins summer 2000 workshop,” Proc. Wks.. Multim. Sig. Process., pp. 619-624, 2001.

Chapter 17

IS OUR DRIVING BEHAVIOR UNIQUE?

Kei Igarashi[1], Kazuya Takeda[1], Fumitada Itakura[1], and Hüseyin Abut[2]

[1] Center for Integrated Acoustic Information Research (CIAIR), Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, JAPAN, http://www.ciair.coe.nagoya-u.ac.jp; [2]ITC, Nagoya University, Japan, Sabanci University, Istanbul, Turkey and ECE Department, San Diego State University San Diego, CA 92182, Email: abut@akhisar.sdsu.edu

Abstract: In this chapter, uniqueness of driver behavior in vehicles and the possibility to use in personal identification has been investigated with the objectives to achieve safer driving, to assist the driver in case of emergencies, and to be part of a multi-mode biometric signature for driver identification. Towards that end, the distributions and the spectra of pressure readings from the accelerator and brake pedals of drivers are measured. We have attempted to use the linear combination of these pedal pressure signals as the feature set. Preliminary results indicate that drivers apply pressure to pedals differently. Are they distinctly unique to be used an independent biometric to identify the individual? Even though our findings at this time are not conclusive, additional features, time-series analysis of the collected data and/or integration these features with audio and video inputs are being investigated.

Keywords: Driving Behavior, biometric signatures, break pedal pressure, acceleration, acceleration pedal pressure, steering wheel angle, data collection vehicle, linear prediction, multi-mode sensors, and Gaussian mixture model.

1. INTRODUCTION

Automated biometric identification is a multidisciplinary scientific field to determine the identity individuals from a set of features based on who they are, what do they posses and how they behave. A number of biometrics has been evaluated in trust building for numerous civic and business transactions, and in forensic authentication applications [1-6,18,19]. These include identification of individuals from their physical features such as fingerprints, hand geometry, face, retina, and iris.

The second class is classified as behavioral signatures, which include voice, style of hand-writing, key-stroke dynamics, motion video, gait, lip-reading, and several others.

Personal identification by digital signatures based on Public Key Infrastructure (PKI), passwords and smart-cards fall into the class of what we posses.

Finally, Deoxyribo Nucleic Acid (DNA) is the one-dimensional ultimate unique code for a person's uniqueness - except for the fact that identical twins have identical DNA patterns. Together with dental records, it has been widely used in personal identification mostly for forensic applications. Since these last two groups do not involve signal processing and they have not been normally studied in the realm of signal processing. Furthermore, they have no applicability in vehicular applications.

Traditionally, features used in identification have been extracted from answers to only one of the three fundamental questions above. Depending on the application, the performance in terms of accuracy and robustness can vary between excellent to unacceptable. In particular, the chamber, where the systems are deployed has been the major deciding factor between the success and failure. For instance, the systems which give excellent results in a controlled testing environment have yielded almost all the time unacceptably poor performance in real-life situations. These include the cockpit, crowded rooms, shopping centers and, in particular, moving vehicles.

Many practical and even costly signal enhancement procedures have been resorted to improve the performance without much success, which in turn, has significantly limited the penetration of biometrics into the realm of e-transactions, i.e., e-business, m-commerce (business in mobile environment) and p-commerce (secure transaction over phone.)

Recently, algorithms using the multi-mode sensor approach to biometric identification have been developed with encouraging results in Chapter 16 and in [10-12]. In particular, the combination of feature sets extracted from iris, finger and video information [10-12]; the fusion of audio and video characteristics in Chapter 16 and the resulting improved performance can be shown as examples in the right direction.

In this paper, we focus on behavioral signals obtained from the driving characteristics of individuals, namely, the distributions and the spectra of pressure readings from the accelerator and brake pedals under various driving conditions. At first, an answer to the question in the title of this paper will investigated:

Is our driving behavior unique? Or equivalently,

Can we use signals obtained from our driving behavior as feature sets in personal identification?

Subsequently, we would like to address the issue of utilization of these behavioral signals for identifying driver behavior with objectives of safer driving, intelligent assistance for road emergencies, and robust communications. Eventually, we hope to develop personal identification with high accuracy and robustness within the framework of a multi-mode e-transaction in cars.

2. IN-CAR DATA COLLECTION

As part of an on-going study on collection and analysis of in-car spoken dialog corpus, 800 drivers have driven a specially equipped vehicle in Nagoya, Japan between 1999 and 2001. Recorded data specifications are listed in Table 17-1, which consists of twelve channels of dialog speech, three channels of video from different angles, the accelerator pedal pressure and brake pedal pressure readings, the vehicle speed in km/h, the engine speed in rpm and the steering angle in degrees. In addition, the location of the vehicle has been recorded every second by a differential GPS device mounted in the vehicle. Detailed information on this corpus study can be found in Chapter 1 and in [14, 16]. In this work, we have utilized only three out of a total of five different vehicle control signals, namely, the accelerator pedal pressure, brake pedal pressure and the vehicle speed in kilometers per hour (km/h). The pressure readings were sampled at 1.0 KHz.

Speech	Sampling: 16 kHz, 16-bit/sample, 12 channels
Video	MPEG-1, 29.97frames per second, 3channels
Control Signals	Acceleration, Accelerator Pedal Pressure, Brake Pedal pressure, Steering Wheel Angle, Engine RPM, Vehicle Speed: Each at 16 bit/sample and 1.0 kHz.
Location	Differential GPS: one reading per second

Table 17-1. Recorded Data Specifications

3. FREQUENCY-DOMAIN ANALYSIS

To avoid the temporal effects, we have decided to study the problem in the frequency-domain with the hopes of extracting feature sets for driver individuality in a precise, robust and consistent manner. Towards that end, we have explored the variations in the long-term spectra of the accelerator pedal and the brake pedal for several drivers, which are illustrated in Figure 17.1. Spectra are computed from the signals over a period of approximately twelve minutes for each driver. As it can be observed that the amplitudes are greater in the low-frequency region, which implies that these pedal pressures tend to change relatively slowly. In spite of significant driver-to-driver differences there is no clear-cut indication of driver individuality form these long-term frequency spectra. We think that the long-term spectra do not take into account the non-stationary characteristics of moving vehicles, traffic, the road conditions, and the driver behavior as response to these. Therefore, we have decided to focus on other signal processing avenues.

4. PEDAL PRESSURE STATISTICS

After observing non-conclusive results from long-term spectral analysis, we have turned our attention to the probability theory by computing the distributions of the accelerator and brake pedal pressures among drivers - both female and male. These are displayed in Figure 17-2. These plots show the relative frequency as a function of pressure readings in kilogram-force per centimeter square (kgf/cm^2) for the accelerator pedal and the break pedal, respectively --1.0 kgf is equal to 9.8 Newtons. It is worth noting that these readings are taken from sensors attached to the pedals.

There are noticeable differences among drivers the way they press each pedal. Their habits in applying pressure to these two pedals in handling a vehicle differ significantly as well. Some drivers accelerate in multiple stages, whereas others tend to press the accelerator in a continuous and smooth manner.

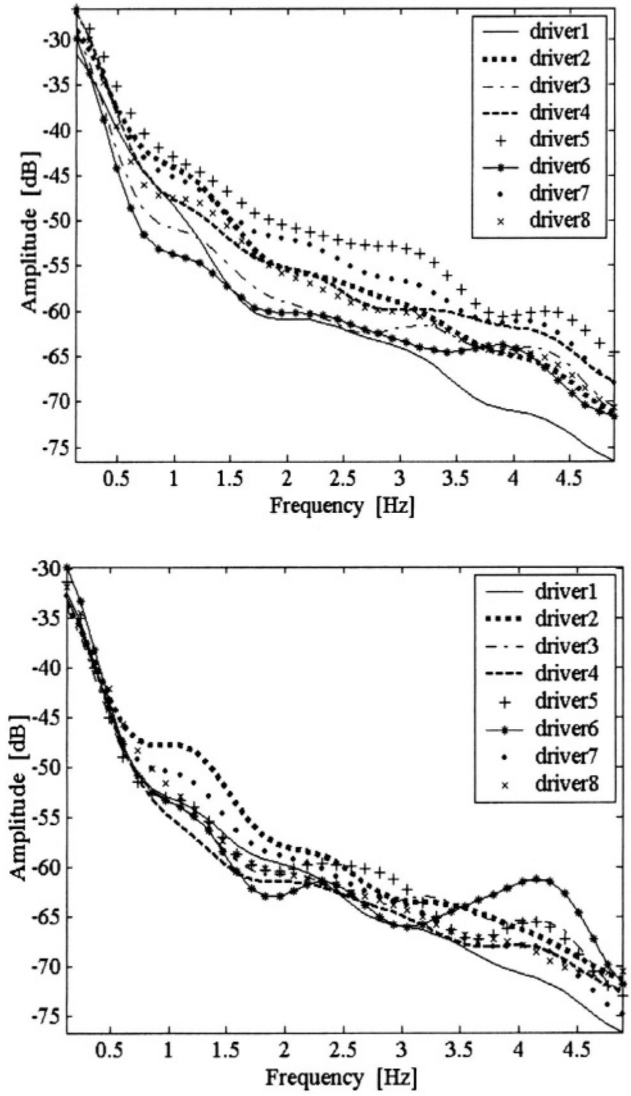


Figure 17-1. Long-term spectra of the accelerator pedal pressure for eight different drivers (top) and that of brake pedal pressure (bottom).

Similarly, the brake pressure application is observed to vary from driver to driver considerably. There are drivers who exhibit a single-step continuous breaking action, an initial big kick in the pedal followed by a number of smaller kicks, and multiple kicks with close values. This can be attributed to the way a particular driver has adjusted himself/herself to best use the vehicle they normally drive.

In particular, the relative frequency of the accelerator pedal pressure is concentrated under 2.0 kgf/cm^2 for driver 1 with a peak at 0.35. However, its brake pressure has sharp peaks around 0.25 and 1.7 kgf/cm^2. The first peak is expectedly the initial impact on the brake pedal after making the decision to stop or to slow down.

On the other hand, driver 3 has multiple peaks over a very long range after the initial impact for the accelerator behavior but it has a sharp peak around 3.9 in the brake pressure plot. Yet another observation is the brake histograms for drivers 2 and 3 are regularly higher that of driver 6.

Despite the apparent variations among these eight drivers, unfortunately, it was not clear from these plots that neither of the two measurements alone would be sufficient to identify the driver completely.

5. INTEGRATION OF MULTI-SENSOR DATA

Limitations imposed by unimodal treatment of driving features could be overcome by using multiple modalities or data fusion as it was recently done in a number biometric systems (Chapter 16 in this book and [10,17]. Preliminary findings from such systems, known as *multimodal biometric systems* indicate higher performance and more reliable due to the presence of multiple, independent pieces of evidence. Data fusion has been effectively used in speech processing community very successfully since 1970s. Excitation signals, gain, zero-crossing rate, pitch information, and LPC coefficients or their offsprings have been fused in one form or another in speech compression, speech/speaker recognition and speaker verification applications. In this section, we propose a multiple sensor version of the ubiquitous linear prediction model for studying the driver individuality.

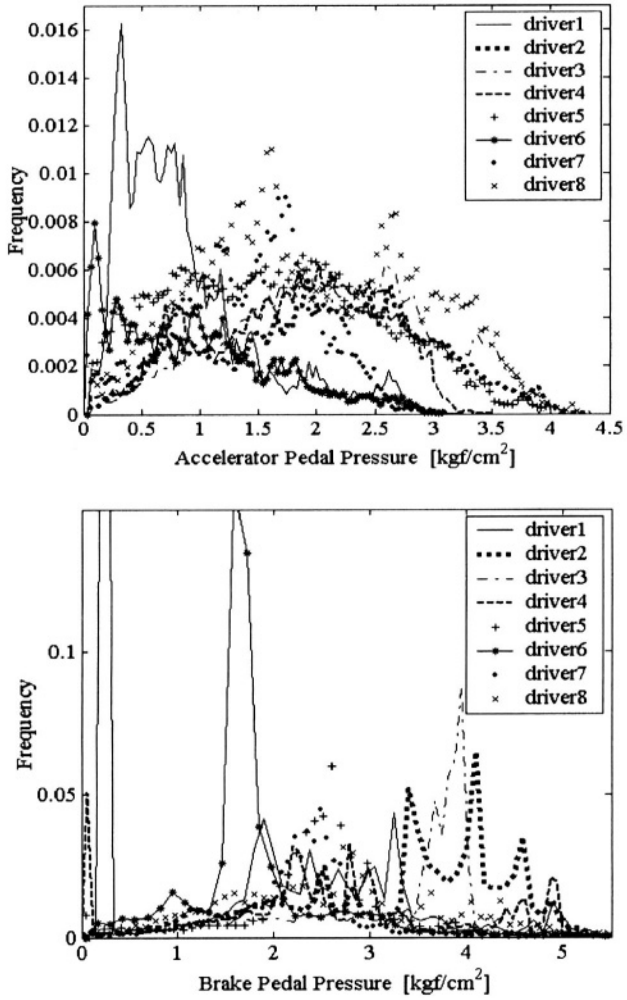


Figure 17-2. Distributions of accelerator pedal pressure (top) and of brake pedal pressure (bottom).

5.1 Combined Observation of Multiple Sensor Data

Time-stamps of the accelerator pedal pressure, the brake pedal pressure, the acceleration itself, and the speed of the vehicle have been plotted in

Figure 17-3. As the accelerator pedal pressure raises, i.e., large positive, the vehicle starts accelerating. On the other hand, as the brake pedal pressure increases, the vehicle slows down with a negative acceleration. Since the drivers can only apply pressure to either the accelerator pedal or the brake pedal, i.e., both feet are not used at the same time, these two signals are mutually exclusive, which is explicitly seen in the plots. By integrating these facts and the significantly different driving tendencies among the collected data, it is quite possible to extract the individuality of drivers using the linear prediction theory.

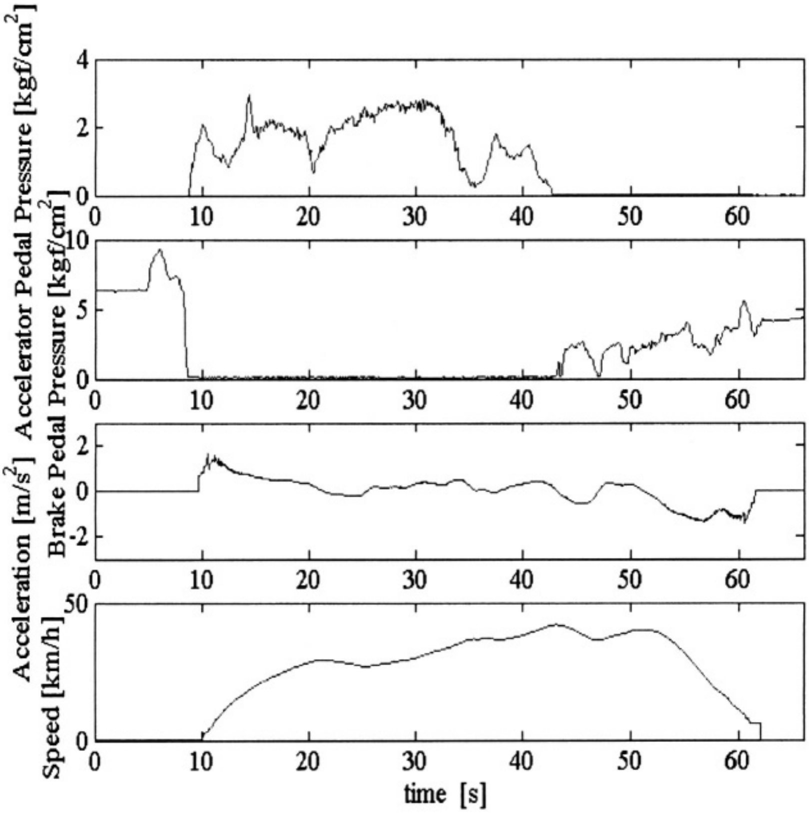


Figure 17-3. Plots of acceleration pressure, the brake pedal pressure and the vehicle speed as a function of time.

5.2 Linear Prediction Model for Driver Behavior

With a goal of extracting individuality from these three measurements and the physical realities of moving vehicles a method based on Linear Prediction (LPC) Theory is proposed. LPC is now a ubiquitous method not only for speech but also other signal processing realms including image processing, geophysics and earthquake studies due to is effectiveness, tractability and computational ease.

At a given discrete time *t,* let us assume that the relation between the acceleration signal x_t and the acceleration pedal pressure y_t, and the brake pedal pressure z_t is given by:

$$x_t + \sum_{i=1}^{P} \alpha_i . x_{t-i} + \sum_{i=1}^{P} \beta_i . y_{t-i} + \sum_{i=1}^{P} \gamma_i . z_{t-i} = \varepsilon_t \tag{1}$$

where ε_t is an uncorrelated random variable with zero mean and variance σ^2. In linear prediction (LPC) theory, the present acceleration value is estimated in terms of its previous values, the associated excitation signals, and the parameter set θ:

$$\theta = \{\alpha_1, \alpha_2, \cdots, \alpha_P, \beta_1, \beta_2, \cdots, \beta_P, \gamma_1, \gamma_2, \cdots, \gamma_P\}$$

where the first group of parameter set forms the weights for the acceleration history, the second and third sets $\{\beta_i\}$ and $\{\gamma_i\}$ are the coefficients for the pressure sensor history for the accelerator and the break pedal, respectively. As expected, ε_t would be the excitation at the time instant *t.* We have thus reformulated the vehicle acceleration behavior as an extended multi-sensory linear prediction problem.

In our case, the optimum parameter set is found by the usual minimization of the total prediction error *E*:

$$E = \sum_{i=1}^{P} \varepsilon_i^2 = \sum_{i=1}^{P} \left[x_t + \sum_{i=1}^{P} \alpha_i . x_{t-i} + \sum_{i=1}^{P} \beta_i . y_{t-i} + \sum_{i=1}^{P} \gamma_i . z_{t-i} \right]^2 \tag{2}$$

We differentiate E with respect each and every parameter in (2) and set to zero:

$$\frac{\partial E}{\partial \alpha_i}=\frac{\partial E}{\partial \beta_i}=\frac{\partial E}{\partial \gamma_i}=0 \quad for\ i=1,2,\cdots P \tag{3}$$

The resulting set of simultaneous equations become:

$$\begin{aligned}&\sum_{k=1}^{P}\alpha_k\cdot\sum_{t=t_0}^{t_1}x_{t-i}.x_{t-k}+\sum_{k=1}^{P}\beta_k\cdot\sum_{t=t_0}^{t_1}x_{t-i}.y_{t-k}\\&\quad+\sum_{k=1}^{P}\lambda_k\cdot\sum_{t=t_0}^{t_1}x_{t-i}.z_{t-k}=-\sum_{t=t_0}^{t_1}x_{t-i}.x_t\end{aligned} \tag{4a}$$

$$\begin{aligned}&\sum_{k=1}^{P}\alpha_k\cdot\sum_{t=t_0}^{t_1}y_{t-i}.x_{t-k}+\sum_{k=1}^{P}\beta_k\cdot\sum_{t=t_0}^{t_1}y_{t-i}.y_{t-k}\\&\quad+\sum_{k=1}^{P}\lambda_k\cdot\sum_{t=t_0}^{t_1}y_{t-i}.z_{t-k}=-\sum_{t=t_0}^{t_1}y_{t-i}.x_t\end{aligned} \tag{4b}$$

$$\begin{aligned}&\sum_{k=1}^{P}\alpha_k\cdot\sum_{t=t_0}^{t_1}z_{t-i}.x_{t-k}+\sum_{k=1}^{P}\beta_k\cdot\sum_{t=t_0}^{t_1}z_{t-i}.y_{t-k}\\&\quad+\sum_{k=1}^{P}\lambda_k\cdot\sum_{t=t_0}^{t_1}z_{t-i}.z_{t-k}=-\sum_{t=t_0}^{t_1}z_{t-i}.x_t\end{aligned} \tag{4c}$$

where $i=1,2,...,P$ and P is the order of prediction in this linear model. Simultaneous solutions of (4a, 4b, 4c) yield the optimum linear feature set for the acceleration signal at time t.

5.3 Multi-Sensor Linear Prediction Experiments

In this set of experiments, we have utilized the data from 84 different drivers. Each driver was observed to make different number of stops and accelerations depending upon the prevailing traffic from the start to the turning off of the engine. We have decided to break the trip into segments.

The term called "period" is used as the basic temporal unit and it is defined as the time elapsed from the instant any pressure applied to the accelerator until the next stop. All together there were a total of 510 periods of data from our driver set. The accelerator pedal pressure, the brake pedal pressure, and the acceleration signal were the inputs to the prediction model as proposed in (1). The acceleration signal is calculated as the simple time gradient of the vehicle speed between two adjacent samples. In the data collection phase, these three pieces of information were digitized at a sampling rate of 1.0 kHz. However, we have down-sampled by a factor of 1:100 resulting at a data rate of 10 Hz in our experiments.

First, it is investigated how the prediction accuracy would change for varying prediction orders *P=1, 2, 4, 8, 16, 32.* The results are displayed in Figure 17.6 and they will be discussed later in this section after identifying four specific cases we have looked into.

Next we have next studied the change in intra-driver prediction error residual characteristics for four different cases:

1. Acceleration only $\{x_{t-i}\}$; i.e., the case where accelerator pedal pressure and the brake pedal pressure are forced to zero.
2. Acceleration and accelerator only $\{x_{t-i}\}$ and $\{y_{t-i}\}$; brake pressure is zero
3. Acceleration and brake only $\{x_{t-i}\}$ and $\{z_{t-i}\}$; accelerator is zero.
4. No term in Equation (1) is forced to zero; i.e., all three terms are presents.

The resulting error defined in (2) is plotted in Figure 17-4 for each of these four different scenarios together with the acceleration signal itself on the top.

Similarly, we have computed the inter-driver residual error signal as shown in Figure 17.5, where $\{x_{t-i}\}$, $\{y_{t-i}\}$, and $\{z_{t-i}\}$ in equation (1) are from one driver, while the LPC parameters $\{\alpha_i, \beta_i, \gamma_i\}$ are from another driver. As in other technique, it is extremely difficult to have a sense for individuality of drivers.

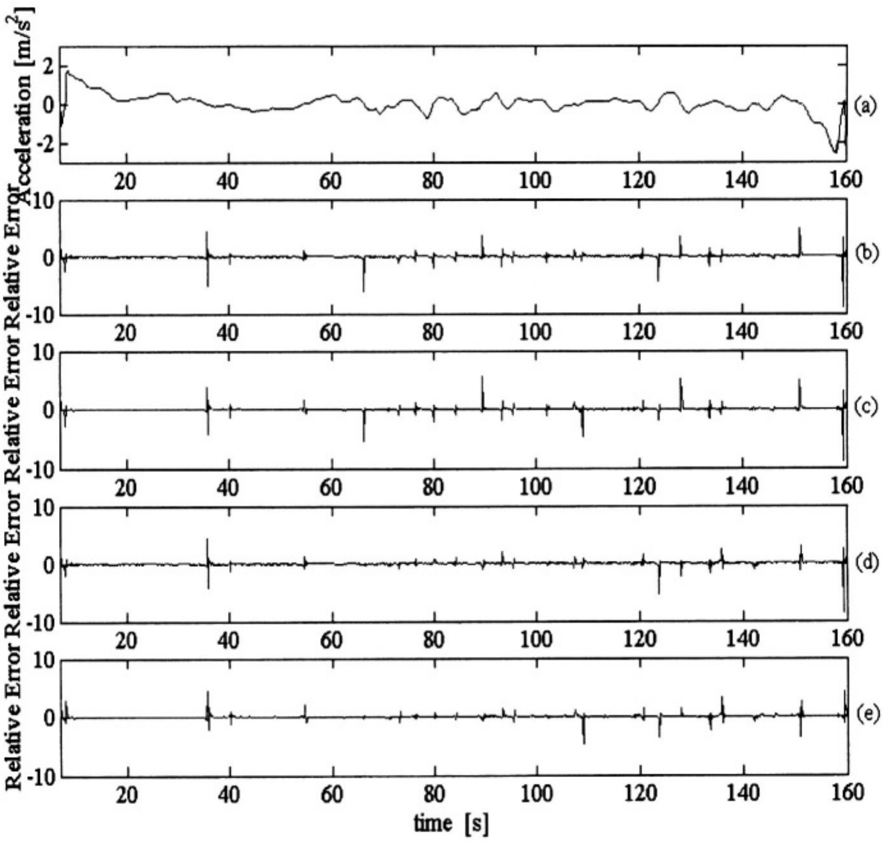


Figure 17-4. Acceleration signal and the intra-driver residuals for four different scenarios.

In Figure 17-6, we have plotted the normalized mean-square error (MSE) as a function of the prediction order P for these four specific cases. The prediction order in the range 20-30 seems to be sufficient. The drop in MSE between cases 2 or 3 and 4 is insignificant. In other words, having all three parameter sets in (1) does not improve the performance; any two results in fairly close results.

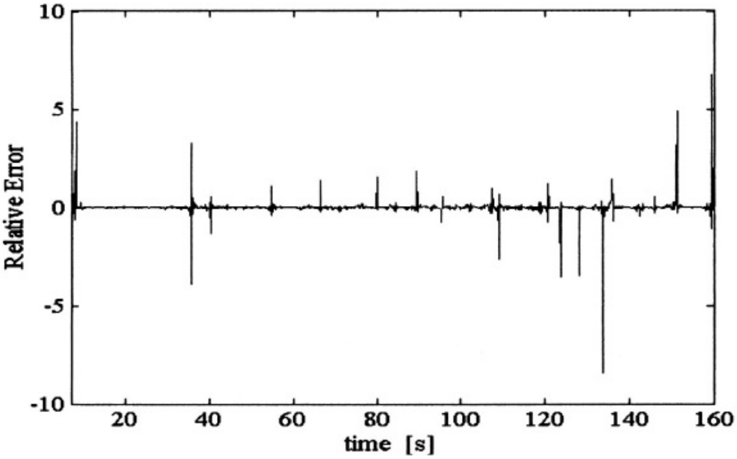


Figure 17-5. The inter-driver residuals signal for Equation (1).

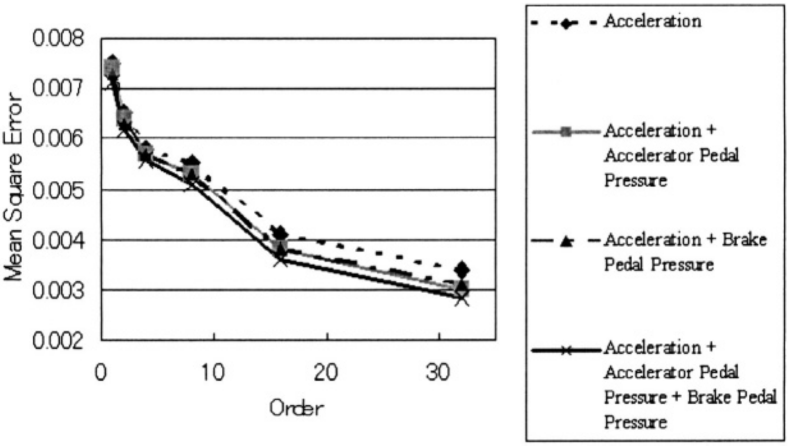


Figure 17-6. Mean-Square Error (MSE) as a function of prediction order P.

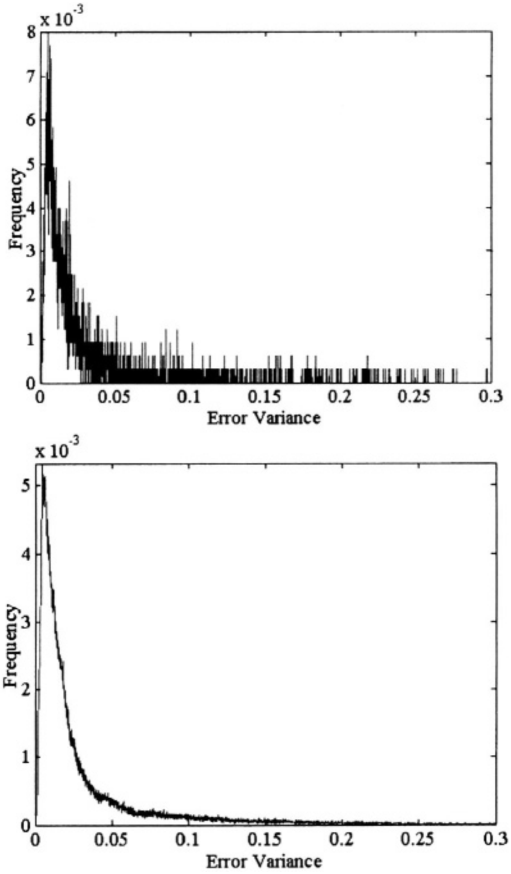


Figure 17-7. Distribution of MSE variance for intra-driver and inter-driver tests.

Finally, we have studied the distribution of the error variance for intra-driver and inter-driver scenarios. These are plotted in Figure 17-7. It is again apparent that there are considerable differences between these two situations. While the dynamics of the intra-driver is higher, the inter-driver curve is very smooth.

6. LESSONS LEARNED AND RECENT EXPERIMENTS

6.1 Lessons Learned

In this study, we have explored the possibility of driver identification from three behavioral signals measured in a data collection vehicle specially designed for construction of an in-car spoken dialog corpus. These were the pressures applied to accelerator and brake pedals and the speed of the vehicle, more precisely, and the acceleration signal. There are a few interesting yet enlightening findings from these costly experiments, traditional spectral and statistical methods, and the proposed extended linear prediction analysis technique:

- There are significant differences among drivers the way they apply pressure to the accelerator and the brake pedal from a probabilistic approach, which can be used in identification when a robust and consistent algorithmic platform is developed.
- Albeit considerable differences in the frequency-domain behavior, there is no simple indication for individuality.
- Dynamics of intra-driver and inter-driver in terms of linear prediction residual are observably different, which again could be very valuable in identifications tasks.
- Linear prediction model as proposed in this chapter has an apparent potential for extracting individuality but it needs to be modified. In particular, the backbone of the LPC approach, i.e., equation (1) does not take one physical fact into consideration: Drivers do not use the brake and accelerator at the same time, where as the model permits that. The curve trajectories of Figures 17-3 and 17-6 clearly support this.
- To remedy this weakness in the model, a switching function, as it is done by a voicing mechanism in the speech processing community, can be incorporated.
- An alternative technique could be to recast the problem within the framework of Kalman Filtering or time-series analysis.

6.2 Recent Experiments

At present, we are investigating two alternative approaches to this problem: First technique is based on correlation filters, which have been very used with encouraging results in multi-sensor biometric identification [2,11]. A similar approached using these two pressure readings, the acceleration signal (vehicle speed), and other behavioral data including the steering wheel information can be developed to better identify the drivers. Experiments are being currently carried out for developing "meaningful and computationally feasible" MACE filters for each driver. The findings will be reported later. It is difficult to present any meaningful quantities as this stage but the promise is much better that the earlier techniques studied above.

In the second technique, however, we are trying to incorporate Gaussian Mixture Models (GMM) for modeling driver behavior. GMM based techniques have resulted in promising results for speaker identification/verification [3, 4, 18, 19]. In our preliminary experiments, we have chosen a small subset of the 800 driver database (30 drivers with equal gender split) and the average length of the driving data was approximately 20 minutes. The first half of the each data was used for modeling the driver and the latter half has been employed for testing the system. We have experimented with 1, 2, 4, 8 Gaussian mixtures and the sum of the log-likelihood was used as the identification measure.

We have obtained a correct identification rate of 73.3 percent using the both the static and dynamic information of accelerator and brake pedal pressure.

7. CONCLUSIONS

After a number of very interesting and yet-not-so-encouraging results from several different approaches, this encouraging preliminary finding (first success story!) is a very important milestone to achieve our goals of safer driving, assisting drivers in road emergencies, and to be part of a multi-mode biometric signature for driver identification. We are planning to present our GMM approach details and the results in the near future.

ACKNOWLEDGEMENT

This work has been supported in part by a Grant-in-Aid for Center of Excellence (COE) Research No. 11CE2005 from the Ministry of Education, Science, Sports and Culture, Japan. The authors would like to acknowledge the members of CIAIR for their enormous contribution and efforts towards the construction of the in-car spoken dialogue corpus.

REFERENCES

[1] W. Shen and R. Khanna, Editors, "Special Issue on Automated Biometrics," Proceedings of the IEEE, Vol. 85, No: 9, September 1997.

[2] H. Abut, "Digitized and Digital Signatures for Biometric Identification," IEEE SP Society DL -2002, http://akhisar.sdsu.edu/abut/biometricsrepository.html

[3] J.L. Douglas, J.C. Junqua, C. Kotropoulos, R. Kuhn, F. Perronnin and I. Pitas, "Recent Advantages in Biometric Person Authentication", Proc. ICASSP, pp.4060-4063, May. 2002.

[4] D.A. Reynolds, 'An Overview of Automatic Speaker Recognition Technology", Proceedings of the IEEE ICASSP, pp.4072-4075, May. 2002, Orlando, FL.

[5] A.K. Jain and A. Ross, "Fingerprint Mosaicking", Proc. ICASSP, pp.4064-4067, May. 2002.

[6] Y. Yamazaki and N. Komatsu, "A Proposal for a Text Indicated Writer Verification Method", IEICE Trans. Fundamentals, vol.E80-A, no.11, pp.2201-2208, Nov. 1997.

[7] V. Chatzis, A.G. Bors, and I. Pitas, "Multimodal Decision-Level Fusion for Person Authentication", IEEE Trans. Systems, Man and Cybernetics, Part A, vol.29, pp.674-680, Nov. 1999.

[8] N. Oliver and A.P. Pentland, "Graphical Models for Driver Behavior Recognition in a SmartCar", Proceedings of IEEE Intl. Conference on Intelligent Vehicles 2000.

[9] O. Nakayama, T. Futami, T. Nakamura, and E.R. Boer, "Development of a Steering Entropy Method for Evaluating driver Workload", in SAE Technical Paper Series, 1999.

[10] A.K. Jain and A. Ross, "Learning User-Specific Parameters in a Multibiometric System," Proc. ICIP, Rochester, N.Y., September 2002.

[11] B.V.K. Vijayakumar, "Correlation Filters for Biometrics," Proc. ICIP2002, Rochester, N.Y., September 2002.

[12] J.L. Wayman, "Digital Signal Processing in Biometric Identification: A Review," Proc. ICIP, Rochester, N.Y., September 2002.

[13] A. Kanak, E. Erzin, Y. Yemez and A. M. Tekalp, "Joint Audio-Video Processing for Robust Biometric Speaker Identification in Car," Chapter 16 in this book.

[14] N. Kawaguchi, S. Matsubara, K. Takeda and F. Itakura, "Multimedia Data Collection of In-Car Speech Communication," Proc. of the 7th European Conference on Speech Communication and Technology (EUROSPEECH2001), pp. 2027--2030, Sep. 2001, Aalborg, Denmark.

[15] N. Kawaguchi, S. Matsubara, I. Kishida, Y. Irie, Y. Yamaguchi, K. Takeda and F. Itakura, "Collection and Analysis of Multi-Layered In-Car Spoken Dialog Corpus," Chapter 1, in this book.

[16] CIAIR URL : http://www.ciair.coe.nagoya-u.ac.jp/

[17] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," IEEE Transactions on Circuits and Systems for Video Technology -Special Issue on Image- and Video-Based Biometrics, August 2003.

[18] A. Reynolds, "Speaker Identification and Verification using Gaussian Mixture Speaker Models," Speech Communication, vol. 17, pp. 91-108, August 1995.

[19] J. P. Campbell, Jr., "Phonetic, Idiolectic, and Acoustic Speaker Recognition," IEEE SP Society DL -2001, http://akhisar.sdsu.edu/abut/biometricsrepository.html.

Chapter 18

ROBUST ASR INSIDE A VEHICLE USING BLIND PROBABILISTIC BASED UNDER-DETERMINED CONVOLUTIVE MIXTURE SEPARATION TECHNIQUE

Shubha Kadambe
HRL Laboratories, LLC, 3011 Malibu Canyon Rd., Malibu, CA 91320, USA;
Email: skadambe@hrl.com

Abstract: Spoken dialogue based information retrieval systems are being used in mobile environments such as cars. However, the car environment is noisy and the user's speech signal gets corrupted due to dynamically changing acoustic environment and the number of interference signals inside the car. The interference signals get mixed with speech signals convolutively due to the chamber impulse response. This tends to degrade the performance of a speech recognition system which is an integral part of a spoken dialogue based information retrieval system. One solution to alleviate this problem is to enhance speech signals such that the recognition accuracy does not degrade much. In this Chapter, we describe a blind source separation technique that would enhance convolutively mixed speech signals by separating the interference signals from the genuine speech. This technique is applicable for under-determined case i.e., the number of microphones is less than the number of signal sources and uses a probabilistic approach in a sparse transformed domain. We have collected speech data inside a car with variable number of interference sources such as wipers on, radio on, A/C on. We have applied our blind convolutive mixture separation technique to enhance the mixed speech signals. We conducted experiments to obtain speech recognition accuracy using with and without enhanced speech signals. For these experiments we used a continuous speech recognizer. Our results indicate 15-35 % improvement in speech recognition accuracy.

Keywords: Blind source separation, convolutive mixture, under determined, signal enhancement, speech recognition accuracy.

1. INTRODUCTION

Spoken dialogue information retrieval applications are becoming popular for mobile users especially, in automobiles. Due to the typical presence of background noise, echoes, and other interfering signals inside a car, speech recognition accuracy reduces significantly. Since it is very hard to know a priori (a) how the acoustic environment inside a car is changing, (b) the number of interfering signals that are present, and finally, (c) how they get mixed at the microphone (sensor), it is not practical to train recognizers for the appropriate range of typical noisy environments. Therefore, it is imperative that the ASR systems are robust to mismatches in training and testing environments. One solution to robustness is speech enhancement based on spectral estimation followed by subtraction. The speech signal enhancement techniques developed so far (a) remove noise by estimating it in the absence of speech (e.g.., [1]) and (b) separate noise i.e., interference signals from the intended signals (e.g., [2]).

In this Chapter, we consider the problem of signal enhancement as the separation of mixed signals (instead of subtracting the noise effect by estimating it) that are received by an array of typically two microphones. For this, it is necessary to apply blind techniques since the nature and the number of signals and the environment which is mixing these signals are not known a priori. Most of the blind techniques developed so far are based on Independent Component Analysis (ICA). These techniques work well when the number of microphones is equal to the number of signals (intended speech signal plus unintended interfering signals). Since it is not practical to know the number of signals present before hand and also this number could be dynamically changing, the techniques based on ICA are not very appropriate in real-life applications. In addition, due to the chamber effect inside a car signals get mixed convolutively. Hence, we need techniques that can separate convolutively mixed signals and work well when the number of microphones is less than the number of signals present. The blind techniques that work when the number of microphones is less than the number of signals is referred to as under-determined Blind Source Separation (BSS). We have developed one such technique using a probabilistic approach in the sparse domain [3].

In this chapter, we apply that technique for signal enhancement. In the next section, an over view of this technique is provided. In section 3, data collection details inside a vehicle are provided. Section 4 provides the details of ASR experiment and the results. In this section the recognition accuracy results obtained using with and without convolutive under-determined BSS technique are compared. In section 5, we summarize and indicate future direction of our research in this area.

2. UNDER-DETERMINED BLIND CONVOLUTIVE MIXTURE SEPARATION

The method of blind source separation (BSS) attempts to estimate the sources or inputs of a mixing system by observing the outputs of the system without knowing how the sources were mixed together (no a priori knowledge of the system) and what the sources are. The BSS is an important problem and has many applications: e.g., interference free wireless communication and robust automatic speech recognition in spoken dialogue systems on mobile platforms. It is worth noting that in this chapter, the BSS is applied within the framework of robust automatic speech recognition problems. There are two cases (i) instantaneous mixture (IM) where the mixing system has no memory and (ii) convolutive mixture where the length of the filters that are used to represent a mixing system is greater than one. Let N be the number of sensors used to observe the source signals and M be the number of sources. Then the IM case can be written in matrix form as:

$$\mathbf{x}(n) = \mathbf{as}(n) + \mathbf{v}(n) \tag{1}$$

where $\mathbf{x}(n)$ is the mixed signal output matrix of size NxK, $\mathbf{s}(n)$ is the matrix of source signals of size MxK, $\mathbf{v}(n)$ is the additive noise matrix of size NxK, n = 1, 2...K are the time samples and **a** is the mixing matrix (mixing system) of size N by M which is represented in terms of angles or directions of arrival of source signals at the sensors i.e., **a** is a function of θ.

The BSS is an easier problem to solve when $N = M$ (finding **a** matrix); several techniques have been developed. However, the BSS is a more difficult problem to solve when $N < M$. In practice it is not possible to know a priori how many sources are present (e.g., in the case of wireless communication the sources correspond to the signals that get reflected from various scatterers such as buildings and noise and in the case of spoken dialogue systems they correspond to other speakers and noise) and they vary dynamically as the environment changes and hence we will not know how many sensors (e.g., antenna elements in the case of wireless communication and microphones in the case of spoken dialogue system) to use so that it is equal to the number of sources to observe the mixed signals. Therefore, BSS when $N < M$ has more practical applications and a more practical problem to solve.

Recently, several authors have shown the feasibility of BSS when *N* is less than *M* for IM case [4-5, 7]. This can be achieved by transforming the sensor (mixed) signals to the time-frequency domain and using the property of sparseness in the transformed domain to help in the estimation of the mixing matrix. After the mixing matrix has been estimated, it is used to estimate the sources where the sources are assumed to be independent and exhibits a Laplacian density in the sparse transformed domain. Note that in all these methods probabilistic techniques have been and a posteriori log probability has been maximized. This maximization with the assumption of independent sources, the sources exhibiting Laplacian density in the sparse transformed domain and additive white Gaussian noise leads to the minimization of L2 and L1 norms. In [4], the mixing matrix is first estimated as mentioned above and then the source signals are separated using this mixing matrix and minimizing the L1 norm. However [7] uses what the authors call "dual update" approach that iteratively refines the estimate of the source and mixing matrix jointly by minimizing L1 and L2 norms. We have extended this to the convolutive mixture in [3] which is reviewed in the following section.

2.1 Probabilistic BSS for underdetermined IM

This section summarizes our previous algorithm described in [7] and generalizes it with some modifications (a) to handle more than 2 mixtures, (b) to robustly estimate the initial mixing matrix and (c) to speed up the iterative "dual update" algorithm.

2.1.1 Review of "dual update" algorithm

Consider the observed signal **x** given in the Eq. (1). The most efficient techniques for the source separation in the case of underdetermined IM are based on probabilistic approach. These approaches mainly correspond to minimizing the negative log of *a posteriori* likelihood function $P(\mathbf{s}|\mathbf{x},\mathbf{a})$ with respect to **s**. Note that the maximization of log a posteriori probability is equivalent to minimizing the negative log posteriori probability. This likelihood function can further be written as

$$P(\mathbf{s}|\mathbf{x},\mathbf{a}) \propto P(\mathbf{x}|\mathbf{a},\mathbf{s})P(\mathbf{a},\mathbf{s}) = P(\mathbf{x}|\mathbf{a},\mathbf{s})P(\mathbf{a})P(\mathbf{s})$$

by applying the Bayes theorem and assuming statistical independence between ***a*** and ***s***. Here, ***P(a)*** and ***P(s)*** correspond to prior probabilities of ***a*** and ***s***, respectively. By applying the negative log operation to $P(\mathbf{s}|\mathbf{x},\mathbf{a})$ we get:

$$-L(\mathbf{s}|\mathbf{x},\mathbf{a}) = -L(\mathbf{x}|\mathbf{a},\mathbf{s}) - L(\mathbf{a}) - L(\mathbf{s})$$

where L corresponds to $log(P())$. The minimization of the negative log likelihood function of $P(\mathbf{s}|\mathbf{x},\mathbf{a})$ then basically corresponds to minimizing $-(L(\mathbf{x}|\mathbf{a},\mathbf{s}) + L(\mathbf{s}))$ with respect to ***s*** since there is no prior information on ***a.*** Since the accuracy of estimated separated source signals ***s*** depends on the accuracy of estimated ***a*** we think that by jointly optimizing the above log likelihood function with respect to both ***a*** and ***s*** (as evidenced by simulation results and as described in [9]) we can separate the sources signals from the observations more efficiently. For this joint optimization, we developed a "dual update" algorithm in [7] that is briefly described below.

Description of joint minimization algorithm – "dual update": For the joint optimization problem, we consider a sparse domain i.e., the domain where most of the coefficients that correspond to non-signals are small (near zero). In other words, the sparse domain is a domain in which signals can be efficiently represented. This has the advantage of reducing the complexity of the problem (i.e., need to deal with sparse matrix compared to full matrix) of separation of mixed signals. Examples of domains where signals can be efficiently represented are Fourier and wavelet. Here we choose Fourier. Note that in this chapter, when we refer to Fourier, we mean short-time Fourier transform and we are not making a specific distinction between Fourier and the short-time Fourier since it is a special case of Fourier i.e., the windowed Fourier. When we compute the Fourier transform we use the fast Fourier transform (FFT) technique. Next, we assume that (a) the source signals are statistically independent to each other (which is not a strong assumption since in practice source signals are statistically independent to each other and researchers commonly make this assumption) and follow Laplacian probability distribution function in the sparse domains (it has been observed that the Fourier and wavelet coefficients do exhibit Laplacian behavior [9]) and, (b) noise **v** is white Gaussian.

As mentioned above, we first transform the mixed signals in to the sparse domain by applying the Fourier transform. We then apply the probabilistic approach of BSS in the sparse domain. The observed mixed signals in the transformed domain can be written as:

$$W(\mathbf{x}) = \mathbf{a}W(\mathbf{s}) + W(\mathbf{v}) \tag{2}$$

where W is the Fourier transform. This has the same form as the mixed observed signals in the time domain (see Eq.(1)). Therefore, without loss of generality, the problem of BSS in the signal domain and in the transformed sparse domain can be considered equivalent. Therefore, the general probabilistic approach mentioned before applies in the transformed sparse domain. However, to get the separated source signals back from the transformed domain to the time domain, we apply the inverse Fourier transform.

We start with the negative log likelihood function i.e., the cost function $L\left(W(\mathbf{s})\middle|W(\mathbf{x}),\mathbf{a}\right)$ in the sparse domain. With the assumption of Laplacianity of source signals in the sparse domain the prior probability:

$$P\left(W(\mathbf{s})\right)=\frac{\lambda}{2}e^{-\lambda\mathbf{c}^{\mathbf{T}}\left|W(\mathbf{s})\right|}\ \text{where}\,\mathbf{c}^{\mathbf{T}}=[1,1,\cdots 1]$$

a unit vector. By applying the “Laplacianity” of signals, “Gaussianity” of noise and no prior information on ***a***, it can be shown that:

$$L\left(W(\mathbf{s})\middle|\mathbf{a},W(\mathbf{x})\right)=\left[\left(W(\mathbf{x})-\mathbf{a}W(\mathbf{s})\right)^{T}\mathbf{R}_{W(\mathbf{v})}^{-1}\left(W(\mathbf{x})-\mathbf{a}W(\mathbf{s})\right)+\lambda c^{T}\left|W(\mathbf{s})\right|\right]$$

where $\mathbf{R}_{W(\mathbf{v})}$ is the noise covariance matrix. (3)

For mathematical simplicity we assume that the noise covariance matrix is an identity matrix. However, the proposed “dual update” approach works for non-Gaussian noise with covariance greater than unity [7]. With unit covariance assumption and re-writing the above equation in terms of $n = 1,2,\cdots K$ we get:

$$L\left(W(\mathbf{s})\middle|W(\mathbf{x}),\mathbf{a}\right)=\sum_{n=1}^{K}\left(W(\mathbf{x}_n)-\mathbf{a}W(\mathbf{s}_n)\right)^{2}+\lambda\mathbf{c}^{\mathbf{T}}\left|W(\mathbf{s}_{n})\right|$$

where $\mathbf{x}_n$ & $\mathbf{s}_n$ are the column vectors of **x** & **s**. (4)

From (4), it can be seen that the first term corresponds to L2 norm where as the second term corresponds to L1 norm. Therefore, our “dual update” approach corresponds to minimizing L2 and L1 norms simultaneously. Note first, we consider the minimization of L2 norm that leads to the estimation of unknown mixing matrix ***a***. For this the above equation is differentiated with respect to ***a*** and set to zero. By doing this we get:

$$\frac{\partial L\left(W(\mathbf{s})\middle|W(\mathbf{x}),\mathbf{a}\right)}{\partial\mathbf{a}}=2\sum_{n=1}^{K}\left(W(\mathbf{x}_n)-\mathbf{a}W(\mathbf{s}_n)\right)^{T}W(\mathbf{s}_n^{i})=0 \qquad (5)$$

Using the individual columns *j* of W(**s**), ***a*** and replacing the summation with the expectation operation, the above equation can be written as:

$$E\left\{W(\mathbf{x}_n)^T W(\mathbf{s}_n^i)\right\} = E\left\{\left(\sum_{j=1}^{M} \mathbf{a}_j^T W(\mathbf{s}_n^j)^T\right) W(\mathbf{s}_n^i)\right\}. \tag{6}$$

By substituting:

$$E\left\{W(\mathbf{s}_n^i)^T W(\mathbf{s}_n^j)\right\} = 0 \text{ for } i \neq j$$

based on the assumption that the source signals are statistically independent in the sparse domain the above equation can be written as:

$$\Sigma_{\mathrm{XS}} = \left[\mathbf{a}_1, \mathbf{a}_2, \cdots \mathbf{a}_M\right]\Sigma_{\mathrm{S}} \text{ where } \Sigma_{\mathrm{S}} \text{ is the } \mathrm{cov}(W(\mathbf{s})) = E\left\{W(\mathbf{s})^T W(\mathbf{s})\right\} \tag{7}$$

and

$$\Sigma_{\mathrm{XS}} = E\left\{W(\mathbf{x})^T W(\mathbf{s})\right\} = \left[\sigma_{\mathrm{Xs}_1}, \sigma_{\mathrm{Xs}_2} \cdots \sigma_{\mathrm{Xs}_N}\right] \tag{8}$$

Then the estimated ***a*** matrix is:

$$\hat{\mathbf{a}} = \left[\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2 \cdots, \hat{\mathbf{a}}_M\right] = \Sigma_{\mathrm{XS}}\Sigma_{\mathrm{S}}^{-1}. \tag{9}$$

There is no closed form solution to minimize both L2 and L1 norm of (4) simultaneously. However, we can solve this system iteratively by applying the Linear Equality Constraints (LEC) optimization technique [10] by noting that (9) can be used as the set of linear constraints. The LEC corresponds to:

$$\begin{aligned} &\text{minimize } \lambda \mathbf{c}^T \left|W(\hat{\mathbf{s}})\right| \\ &\text{subject to } \hat{\mathbf{a}} = \Sigma_{\mathrm{XS}}\Sigma_{\mathrm{S}}^{-1} \end{aligned} \tag{10}$$

The LEC in essence corresponds to finding $W(\hat{\mathbf{s}})$ under the linearity constraint such that the $\left|W(\hat{\mathbf{s}})\right|$ (L1 norm) is minimized. The LEC optimization problem can be solved by applying the line search together with the projection gradient method. One of the ways to find the lines or direction of lines is by applying Armijo rules of line search. We applied this technique.

In short, by applying the above described LEC, we can solve our problem, i.e., the minimization of (4) iteratively by using the following two steps:

1. Find W($\hat{\mathbf{s}}$) that min $\lambda\mathbf{c}^T|W(\hat{\mathbf{s}})|$ (This corresponds to minimizing $|W(\hat{\mathbf{s}})|$ under the linear constraint: $\hat{\mathbf{a}} = \Sigma_{\mathrm{XS}}\Sigma_{\mathrm{S}}^{-1}$.)
2. Use W($\hat{\mathbf{s}}$) from Step 1 and estimate $\hat{\mathbf{a}} = \Sigma_{\mathrm{XS}}\Sigma_{\mathrm{S}}^{-1}$ (This corresponds to finding lines or directions of lines.)

From the above set of equations, it can be seen that to get a fairly good initial estimate of s i.e. $\hat{\mathbf{s}}$ that is used in the step 1 to begin the iterative process, a good initialization of $\boldsymbol{a}$ is needed. Note that even though splitting of the minimization of Eq. (4) into two parts as described above is not theoretically justified since $\boldsymbol{a}$ is not convex, however, we have found that given a good initial estimate of $\boldsymbol{a}$, the "dual update" algorithm converges fast and results in accurate final estimation of $\boldsymbol{a}$ and $\boldsymbol{s}$. For the initial estimate of $\boldsymbol{a}$ an information theoretic based method was used which is described in detail in [7].

Note that the "dual update" approach described above is not a single maximum a posteriori estimation (MAP) of $\boldsymbol{a}$. However, it corresponds to much more tractable joint MAP of $\boldsymbol{a}$ and s.

It is important to note that the initialization of $\boldsymbol{a}$ matrix should not be confused with the classical approach of single MAP estimate of $\boldsymbol{a}$ and the estimation of separated source signals by inverting the estimated $\boldsymbol{a}$. Instead, the initialization is only for a good starting point for the iterative dual update algorithm.

To summarize, the steps of our "dual update" algorithm are:

1. Find $W(\hat{\mathbf{s}})$ that minimizes $\lambda\mathbf{c}^T|W(\hat{\mathbf{s}})|$ under the linear constraint $\hat{\mathbf{a}}\Sigma_{\mathrm{S}} = \Sigma_{\mathrm{XS}}$
2. Use $W(\hat{\mathbf{s}})$ from Step 1 to create a new estimate of the mixing matrix $\hat{\mathbf{a}} = \Sigma_{\mathrm{XS}}\Sigma_{\mathrm{S}}^{-1}$
3. Repeat Steps 1 and 2 until a convergence or stopping criterion is met.

We start our "dual update" algorithm with an initial estimate of the mixing matrix obtained using the technique based on mutual information and angle thresholding technique described in [7].

2.1.2 Generalization of the "dual update" algorithm for IM

By representing the Fourier (short time Fourier) transform W(**x**) = **X**(k,m), W(**s**) = **S**(k,m) and W(v) = **V**(k,m) which are the time-frequency representations of $\mathbf{x}(n)$, $\mathbf{s}(n)$ and $\mathbf{v}(n)$, respectively, and $W(\mathbf{a}) = \mathbf{A}(k,m)$ equation (2) can be re-written as

$$\mathbf{X}(k,m) = \mathbf{A}(k,m)\mathbf{S}(k,m) + \mathbf{V}(k,m) \tag{11}$$

Since $\mathbf{a}$ represents an instantaneous system, $\mathbf{A}(k,m) = \mathbf{A}(k) = \mathbf{A}$ and is constant for all frequency bands k and time m.

In order to find an initial estimate of $\mathbf{A}$ the mutual information between sensors is computed for each sub-band. The sub-band with the maximum mutual information is chosen since it represents a band that exhibits the most separation and hence, can best estimate initial $\mathbf{a}$. For this estimation the following equation is used:

$$\theta_{i,j}(k,n) = \arctan\left(\frac{X_j(k,n)}{X_i(k,n)}\right) \tag{12}$$

where k is the chosen sub-band and i and j represent the signal received at the i^{th} and j^{th} sensors. If $S_l(k,n)$ is much larger than all the other at a particular k and n then equation (12) simplifies to:

$$\theta_{i,j}(k,n) \approx \arctan\left(\frac{A_{j,l}S_l(k,n)}{A_{i,l}S_l(k,n)}\right) = \arctan\left(\frac{A_{j,l}}{A_{i,l}}\right) i = 1,..N, j = 1,..i-1 \tag{13}$$

This results in clusters of measurements that correspond to the arctangent of the ratios of rows of Aj and Ai. Several methods could be used to find these clusters. Some authors use peak picking of the histogram [7,11] and others use potential function [4]. The peak picking of the histogram has the disadvantage of difficulty in accurately picking the local maxima.

On the other hand, we use here a hierarchical clustering approach where each observation is taken in succession and merged with nearest neighbor. This is computationally less intensive than finding the two observations that are closest together and then merging them. The result of hierarchical clustering is used as an initial guess for the k-means clustering algorithm. The means of the clusters that are obtained when the k-means clustering algorithm converged are then used for the initial estimate of the mixing matrix. The number of clusters is used as the estimate of the number of sources to be separated. It was empirically observed that the combination of hierarchical and k-means clustering algorithms works better than using either the hierarchical or k-means clustering algorithms individually.

The next step is to jointly minimize the cost function ("dual update"). In this case the sources are modelled as a Laplacian and the noise is assumed to be white Gaussian as mentioned before, so the resulting cost function - log likelihood *L* is:

$$L(\mathbf{S}|\mathbf{A},\mathbf{X}) = \left[(\mathbf{X} - \mathbf{AS}))^T(\mathbf{X} - \mathbf{AS}) + \lambda\mathbf{c}^T|\mathbf{S}|\right] \tag{14}$$

with the assumption that the noise covariance matrix is a unity matrix. Here $\mathbf{c}^{\mathbf{T}} = [1,1,\cdots 1]$, *T* denotes the Hermitian transpose of a matrix and indices k and *m* are not specifically mentioned to simplify the expression. This expression is optimized by first finding $\mathbf{S}(k,m)$ that minimizes $\lambda\mathbf{c}^{\mathrm{T}}|\mathbf{S}(k,m)|$ under the constraint that:

$$\mathbf{X}(k,m)=\mathbf{A}(k)\mathbf{S}(k,m).$$

The second part of the procedure re-estimates **A**(k) so that the sources will be more independent.

The easiest way to perform the first part of the "dual-update" method is to recognize that there is a local minimum whenever there are *N* – *M* zeros in the **S***(k,n)* vector. This can be shown by using a geometrical argument. First we draw the shape formed by all points at a certain cost ε. The resulting shape is an *N*-dimensional cube with vertices located on the axes at a distance ε away from the origin. Now the constraint has dimension *N* – *M*. So, when there is one more source than the sensor the constraint is a line. If the line goes through the cube then the portion inside the cube is at a lower cost and the portion outside is at a higher cost.

If the cube is shrunk until the constraint only touches the edge of the cube, the point of intersection is the lowest cost. If the line is parallel to one of the sides of the box, then there are an infinite number of solutions. This case corresponds to **A** matrix having at least two identical column vectors. The other case requires that the line intersect a vertex of the cube. Of course this occurs when the line passes through a plane created by all combinations of M axes or in other words there are N – M zeros in S(k,n), which yields a finite number of points to check. The point with the lowest cost is the global minimum. Inclusion of this geometric constrained based search not only has speeded up our original "dual update" algorithm and also has generalized it to handle more than two sensors.

2.2 Challenges for the convolutive case

This section focuses on the problems to be addressed when using the above mentioned method ("basic method") to separate signals in the convolutive case where the mixing matrix is not constant as in the case of IM but is a function of time. The logical extension of the "basic method" would be to take the Fourier transform of the signal with an FTT length that is long enough to ensure that the convolution can be approximated as multiplication in the frequency domain. Then the "dual update" algorithm can be applied in each subband independently.

This approach does have several drawbacks. First of all, the algorithm finds the signal separation within an arbitrary scale factor and arbitrary permutation. This means that the scale factors and permutations will need to be consistent between different subbands. Incorrect scale factors cause spectral distortion.

Currently, there is no good method that can come up with consistent scale factors for all the bands. However, the solution adopted in this study is to constrain the mixing system's filter structure such that:

$$\left\|\mathbf{A}_{\mathbf{j}}(k)\right\|^{2} = 1 \tag{15}$$

where $\mathbf{A}_j(k)$ is the j^{th} column vector of $\mathbf{A}(k)$. This was also used in [6].

Permutation estimation: For finding the correct permutation between bands several methods have been developed which are detailed and compared in [12]. Here, we use the inter frequency correlation. The inter frequency correlation relies on the non-stationarity of the sources [8]. It has been shown that for non-stationary signals, adjacent sub-bands are correlated. This can be used in the following equation:

$$\hat{\mathbf{P}}(k) = \underset{\mathbf{P}(k)}{\arg\max} = \sum_{n=1}^{K} \sum_{j=1}^{k-1} \left(\mathbf{P}(k)\overline{\mathbf{S}}(k,n)\right)^{T} \overline{\mathbf{S}}(j,n) \tag{16}$$

where $\mathbf{P}(k)$ is the permutation matrix and $\overline{\mathbf{S}}(k,n)$ is the envelope of signal $\mathbf{S}(k,n)$. The envelope signal is created by passing the absolute values of the source signals through a low pass filter. The permutation of the first subband is designated as the correct permutation. The permutation of the next sub-band is estimated by using (16). The source signals at that subband are permuted according to the resulting $\mathbf{P}(k)$. This is continued for all sub-bands.

Modification of initial estimate of A for a complex case: Next, in the case of convolutive mixture **A**(*k*) is complex. So the initial estimate of the mixing matrix described in section 2.1 needs to be modified. It is modified as follows. The ratio in polar coordinates of the i^{th} and j^{th} rows for the n^{th} column of A is:

$$\frac{A_{j,n}}{A_{i,n}} = \frac{\left|A_{j,n}\right|}{\left|A_{i,n}\right|} e^{\sqrt{-1}\left(\angle A_{j,n} - \angle A_{i,n}\right)} \tag{17}$$

Using an argument similar to Equation (12) that $S_l(k,n)$ is larger than all other sources results in:

$$\begin{aligned} \phi_{i,j} &= \angle X_j(k,n) - \angle X_i(k,n) \\ &= \left(\angle S_l(k,n) + \angle A_{j,;}(k)\right) - \left(\angle S_l(k,n) + \angle A_{i,;}(k)\right) \\ &= \angle A_{j,l}(k) - \angle A_{i,l}(k) \end{aligned} \tag{18}$$

where ϕ is the phase difference and $\angle$ represent the angle operator. This shows that the estimation of **A**(*k*) requires two components – the ratio of magnitudes of **A**(*k*) elements to obtain θ and the difference in phase between the elements to obtain ϕ.

The remaining procedure is same as before as described in section 2.1 in that the clustering approach is used to determine the initial estimate of **A**(*k*). Since ϕ is between 0 and 2π and θ is between 0 and π/2, ϕ is appropriately weighted in the clustering so that the same amount of weight will be placed on the ϕ components as the θ components. A value of ϕ that is slightly larger than 0 should be considered closed to a value that is slightly less than 2π. If the phase difference is close to 0 or 2π then the clustering algorithm could see two clusters. In order to avoid this possibility the histogram of ϕ is computed and the values are shifted so that the discontinuity will occur at a point that would not divide a cluster.

An example scatter plot of ϕ versus θ is shown in Figure 18-1. From this figure it can be seen that three clusters corresponding to three sources are formed without much overlap whose mean values are pretty close to the true values (circled x). Unfortunately, due to the ambiguity in the scale factor the actual phase values of the mixing matrix cannot be determined. However, the use of the phase difference ϕ greatly improves the robustness of the separation for convolutive mixtures and complex IM.

Lastly, the "dual update" algorithm for underdetermined IM chooses a particular frequency sub-band by using mutual information measure to estimate **a.** However, now **A**(k) is no longer constant for all k. This means that each sub-band has to be used. Unfortunately, the separation will perform better in some bands than the others. This is a reality that cannot be escaped. In summary, the steps involved in the BSS of underdetermined convolutive mixture are:

1. Compute the FT (short) of the observed mixtures.
2. For each frequency sub-band
   (a) Obtain an initial estimate of **A** matrix using the procedure described above.
   (b)Apply the "dual update" algorithm iteratively to refine the estimates of **A** and **S**.
3. Find the appropriate permutation of sources in each frequency band using the final estimates of **A** and **S** that are obtained after the "dual update" algorithm converged.
4. Obtain the separated source signals and the final estimate of **a** by applying the inverse FT.

A block diagram of the proposed algorithm for BSS of underdetermined convolutive mixtures is provided in Figure 18-2.

3. IN-VEHICLE DATA COLLECTION DETAILS

A linear microphone array of five microphones was built by CSLR, University of Colorado and was used to collect in-vehicle speech data (see Chapter 2). This array was placed on the visor of the driver side. Another linear microphone array built by Andrea Electronics was placed on the visor of the front passenger.

A reference microphone was placed behind the driver seat facing the rear passenger. Using an eight channel digital audio recording device, seven channels that correspond to five microphones of the linear array, one output of the Andrea microphone array (separate microphone outputs are not available in the case of Andrea microphone array) and one reference microphone data was recorded. A Sport Utility Vehicle (SUV) was employed for this data collection task. Navigation related speech data including the phrases "how far is the airport from Malibu?", "Can you give me directions to the airport?", etc. was collected in side the vehicle for two speakers (one male and one female) under three conditions - quiet (window was up, radio and A/C were off and there was no presence of cross talk), radio on and cross talk present.

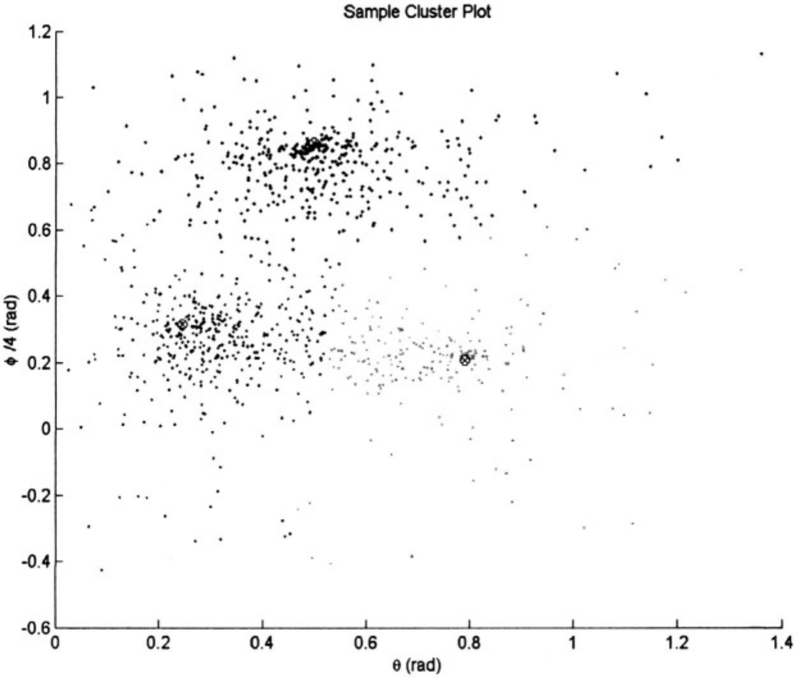


Figure 18-1. Scatter plot of data for BSS of three sources using 2 sensors. The circled x shows the true value computed from the mixing matrix.

Fifty one utterances per speaker for each condition were collected. While digitally recording this data it was sampled at 44.0 KHz. This data was then down loaded to a computer in .wav format and was transcribed orthographically. The speech data was downsampled to 8.0 KHz since the segment based continuous speech recognizer that we used in our experiments expects the data to be sampled at 8.0 KHz.

4. EXPERIMENTS

Speech recognition performance in terms of word recognition accuracy percentage was obtained using the database both using the blind convolutive mixture separation algorithm proposed above and without. An example of a mixed speech signal from two channels and separated four speech signals from the mixed signals using our approach is provided below in Figures 18-3 and 18-4, respectively. From this, it can be seen that our algorithm separated all four speech signals fairly well. The speech recognizer that we have used in this study is a segment based continuous speech recognizer with a vocabulary size of 3000 words. Each segment and segment boundaries are modelled using Gaussian mixture model (GMM). Duration models were also used.

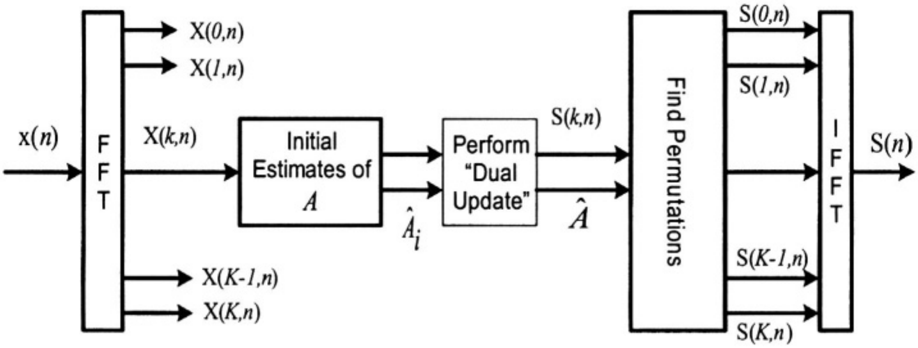


Figure 18-2. The block diagram of the proposed algorithm for BSS of convolutive mixture in the underdetermined case.

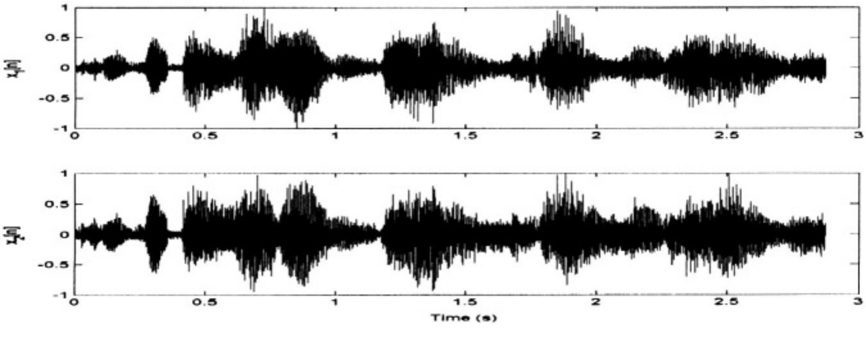


Figure 18-3. Mixed speech signals from two channels.

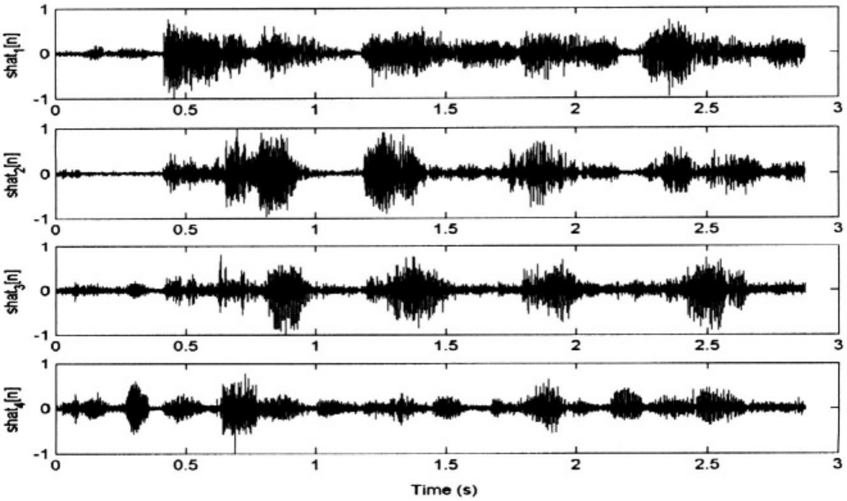


Figure 18-4. Separated four speech signals from two mixtures after applying our algorithm.

While applying the proposed convolutive mixture separation algorithm, the speech data from two channels out of seven - one corresponding to channel 1 of our microphone array and the second corresponding to Andrea microphone array was considered. First, the speech data from channel 1 of our microphone array was used to test the speech recognizer. Next, the speech data that was enhanced by applying the mixture separation algorithm was used. In the Table 18-1 word recognition accuracy for all the three conditions - quiet, radio on and cross talk present with and without enhancing speech signals is provided. From this table, it can be seen that there is a significant improvement in speech recognition accuracy in all three cases. In the case of quiet even though other sources such as radio, cross talk was not present but due to the presence of car engine noise and road noise the speech recognition accuracy degraded; however, when the mixture separation algorithm was used to enhance the speech signals, the accuracy improved significantly (15 %). Note that the number of utterances used for each case is 102.

Operating conditions	Word recognition % Without enhancement	Word recognition % With enhancement
Quiet Moving vehicle but Radio & A/C are off, window up, No cross talk present)	70 %	85 %
Radio on Moving vehicle with Radio on, A/C off, window up, no Cross talk present.	51.5 %	81.4 %
Cross talk Moving vehicle, radio & A/C off, window up, other passengers talking.	45.3 %	80. 2 %

Table 18.1. Speech recognition performance under different conditions with and without the proposed speech enhancement technique.

5. CONCLUSIONS

In this chapter, for robust automatic speech recognition (ASR) inside a vehicle (car), a speech enhancement technique based on blind separation of convolutively mixed signals is applied. This technique is applicable for under-determined case and hence, is a more practical approach to use in real applications such as RASR inside a car as compared to other BSS techniques that work well when the number of sources is equal to the number of sensors. The signal enhancement capabilities of this technique are verified using a measure of improvement in speech recognition accuracy. Our preliminary recognition results of navigation related speech data that was collected in an SUV show that a significant improvement in speech recognition accuracy - 15 to 35% can be obtained by using our blind convolutive mixture separation algorithm. Future work warrants testing of the proposed technique using a larger data set such as the in-vehicle speech data collected by CSLR, Colorado University (see Chapter 2). Also, the performance of our blind convolutive mixture separation can be improved if adaptive beamforming and mixture separation is combined. We are currently working on this. Future work also warrants using this combined beamforming and blind source separation based signal enhancement approach to further improve the speech recognition performance.

REFERENCES

[1] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions of Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, July 2001.

[2] N. Grbic, X-J. Tao, S. E. Nordholm, and I. Claesson, "Blind signal separation using over-complete subband representation," *IEEE Transactions of Speech and Audio Processing,* vol. 9, no. 5, pp. 524–533, July 2001.

[3] J. M. Peterson and S. Kadambe, "A probabilistic approach for blind source separation of underdetermined convolutive mixtures," in *Proceedings ofthe ICASSP,* 2003.

[4] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Processing,* pp. 2353–2362, 2001.

[5] Te-Won Lee, Michael S. Lewicki, Mark Girolami, and Terrence J. Sejnowski, "Blind source separation of more sources than mixtures using overcomplete representations," *IEEE Signal Processing Letters,* vol. 6, no. 4, April 1999.

[6] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources," *IEEE Transactions of Speech and Audio Processing,* vol. 8, no. 3, May 2000.

[7] A. Ossadtchi and S. Kadambe, "Over-complete blind source separation by applying sparse decomposition and information theoretic based probabilistic approach," in *Proceedings of the ICASSP,* 2000.

[8] F. Asano and S. Ikeda, "Evaluation and real-time implementation of blind source separation system using time-delayed decorrelation," in *Proc. ICA,* Helsinki, 2000.

[9] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1," In Vision Research, vol. 37, pp. 3311-3325, 1997.

[10] P. E. Gill, W. Murray and M. H. Wright, Practical Optimization, Chapter 3, Academic Press, 1981

[11] A. Prieto, B. Prieto, C. G. Puntonet, A. Canas and P. Martin-Smith, "Geometric separation of linear mixtures of sources: application to speech signals," Proceedings of the ICA'99. pp. 295-300, January 1999.

[12] D. Kolossa, B. Kohler, M. Conrath and R. Oreglmeister, "Optimal permutation correction by multi-objective genetic algorithms," in Proceedings of ICA, San Diego, CA 2001.

Chapter 19

IN-CAR SPEECH RECOGNITION USING DISTRIBUTED MICROPHONES

Tetsuya Shinde[1], Kazuya Takeda[2], Fumitada Itakura[1]

[1]*Graduate School of Engineering;* [2]*Graduate School of Information Science Nagoya University, 1 Furo-cho, Nagoya 464-8603 Japan* *Email: takeda@is.nagoya-u.ac.jp*

Abstract In this paper, we describe a method for multichannel noisy speech recognition that can adapt to various in-car noise situations during driving. Our proposed technique enables us to estimate the log spectrum of speech at a close-talking microphone based on the multiple regression of the log spectra (MRLS) of noisy signals captured by a set of distributed microphones. Through clustering of the spatial noise distributions under various driving conditions, the regression weights for MRLS are effectively adapted to the driving conditions. The experimental evaluation shows an average error rate reduction of 43 % in isolated word recognition under 15 different driving conditions.

Keywords: In-car-ASR, multiple microphone, linear regression

1. INTRODUCTION

Array-microphone signal processing is known for sometime now to be effective for spatially selective signal capture and, in particular, noisy speech recognition when the locations of the speaker and noise sources are predetermined. However, when the spatial configuration of the speaker and noise sources are unknown or they change continuously, it is not easy to steer the directivity adaptively to the new conditions [1], [2], [3].

Previously, we have proposed multiple regression of log spectra (MRLS) to improve the robustness in the case of a small perturbation of the spatial distribution of the source and noise signals. In that study, log spectra of the signals captured by distributed microphones have been used to approximate that of the close-talking microphone, through linear regression [4]. In addition, we have employed MRLS technique for speech recognition in vehicles and have shown its effectiveness in improving the accuracy of noisy speech recognition. Through the experiments, we also found that further improvement of the recognition accuracy can be achieved if the regression weights are trained for each speaker and/or a particular in-car sound condition that is mainly set the vehicle itself. These scenarios include the background music from the audio system, windows are closed or open, noise from fan/A.C., and the speed of the vehicle. It is worth noting that the computation of regression weights regression weights for a given speaker at enrolment is not difficult. Whereas, changing the weights in order to adapt to the driving conditions is not easy.

The aim of this study is to improve the MRLS so that regression weights can be changed adaptively to the in-car noise conditions. For this purpose, we attempt to benefit from distributed microphones for capturing the spatial distribution of noise sounds.

The rest of the paper is arranged as follows. First, in Section 2, we describe the in-car speech corpus recorded using distributed microphones. The basic idea of MRLS and its extension to the adaptive method are described in Section 3 and Section 4, respectively. In Section 5, experimental evaluations and their results are discussed. Section 6 is a summary of this paper.

2. MULTIPLE REGRESSION OF LOG SPECTRA

2.1 Two-dimensional Taylor-expansion of log-spectrum

Assume that speech signal $x_i(t)$ at i^{th} microphone position is give by a mixture of the source speech $s(t)$ and the noise $n(t)$ convolved with the transfer functions to the position, $h_i(t)$ and $g_i(t)$, i.e.,

$$x_i(t) = h_i(t) * s(t) + g_i(t) * n(t),$$

as shown in Figure 19-1. Assume also that the power spectrum of $x_i(t)$ is given by the 'power sum' of the filtered speech and noise, i.e.,

$$X_i(\omega) = |H_i(\omega)|^2 S(\omega) + |G_i(\omega)|^2 N(\omega),$$

where $S(\omega)$, $X_i(\omega)$ and $N(\omega)$ are the power spectra of the speech signal at its source position, the noisy speech signal at i^{th} microphone position and the noise signal at its source position, respectively. (The frequency index (ω) will be omitted in the rest of paper.) Consequently, the corresponding log-power-spectrum of the signals at the i^{th} microphone positions are given by

$$\log X_i = \log\left\{|H_i|^2 S + |G_i|^2 N\right\}.$$

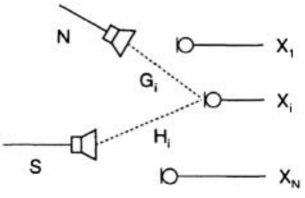


Figure 19-1. Signal captured through distributed microphones.

The derivative of $\log X_i$ can be calculated by

$$\begin{aligned}\Delta \log X_i &= \frac{\partial \log X_i}{\partial \log S}\Delta \log S + \frac{\partial \log X_i}{\partial \log N}\Delta \log N \\ &= a_i \Delta \log S + b_i \Delta \log N.\end{aligned} \tag{1}$$

Where a_i and b_i are given by

$$\begin{aligned}a_i &= \frac{|H_i|^2 S}{|H_i|^2 S + |G_i|^2 N} \\ b_i &= \frac{|G_i|^2 N}{|H_i|^2 S + |G_i|^2 N}.\end{aligned} \tag{2}$$

Note that both a_i and b_i are the functions of the ratio between signal and noise at their source positions, i.e., *S/N*. Small deviations of the log-power-spectrum of the signal at the i^{th} microphone position can be approximated by a two-dimensional Taylor series expansion around X_i^0, i.e.,

$$\log X_i - \log X_i^0 \approx a_i(\log S - \log S^0) + b_i(\log N - \log N^0), \tag{3}$$

where

$$\log X_i^0 = a_i \log S^0 + b_i \log N^0.$$

Using superscript $(\bullet)^{(d)}$ for the deviation from $(\bullet)^0$, e.g. $\log X_i^d = \log X_i - \log X_i^0$, the Taylor expansion can be rewritten by

$$\log X_i^{(d)} \approx a_i \log S^{(d)} + b_i \log N^{(d)}. \qquad (4)$$

2.2 Multiple regression of multi-channel signals

Approximation of $\log S^{(d)}$ by the multiple-regression of $\log X_i^{(d)}$ has the form

$$\log S^{(d)} \approx \sum_{i=1}^{N} \lambda_i \log X_i^{(d)}.$$

By substituting equation (4), the regression error of the approximation, ϵ, can be calculated as follows.

$$\begin{aligned} \varepsilon &= \left[\log S^{(d)} - \sum_{i=1}^{N} \lambda_i \log X_i^{(d)}\right]^2 \\ &= \left[\log S^{(d)} - \sum_{i=1}^{N} \lambda_i \left\{a_i \log S^{(d)} + b_i \log N^{(d)}\right\}\right]^2 \\ &= \left[\left(1 - \sum_{i=1}^{N} \lambda_i a_i\right) \log S^{(d)} - \sum_{i=1}^{N} \lambda_i b_i \log N^{(d)}\right]^2 \end{aligned}$$

Assuming the orthogonality between $\log S^{(d)}$ and $\log N^{(d)}$, the expectation value of the regression error becomes

$$E\left[\left(1 - \sum_{i=1}^{N} \lambda_i a_i\right)^2 \{\log S^{(d)}\}^2 + \left\{\sum_{i=1}^{N} \lambda_i b_i\right\}^2 \{\log N^{(d)}\}^2\right].$$

The minimum regression error is then achieved when

$$\sum_{i=1}^{N} E\{a_i\} \lambda_i = 1, \qquad \sum_{i=1}^{N} E\{b_i\} \lambda_i = 0.$$

Thus, the optimal $\{\lambda_i\}$ can be uniquely determined as a vector that is orthogonal to $\{b_i\}$ and its inner product with $\{a_i\}$ is equal to unity. The relationship among these three vectors are shown in Figure 19-2.

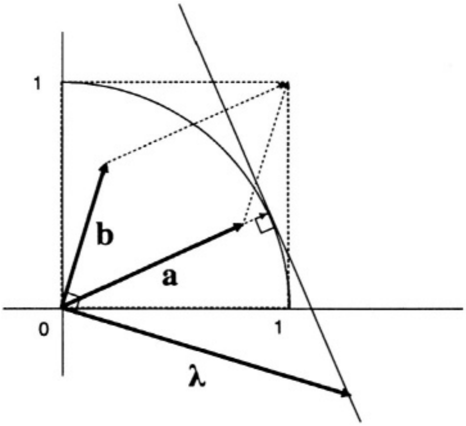


Figure 19-2. The geometric relationship among optimal regression weights λ and the Taylor series coefficients a and b. In the log-power-spectrum domain, $a_i + b_i = 1$ continues to hold.

a_i and b_i correspond to the Signal-to-Noise and Noise-to-Signal ratios, respectively, at the microphone position, and the relationship

$$a_i + b_i = 1$$

holds for every microphone position. Therefore, once λ_i is given, both a_i and b_i are uniquely determined. Multiple regression on the log-power-spectrum domain can be regarded as an implicit estimation of the local SNR at each microphone position.

On the other hand, when multiple-regression is performed on the power-spectrum domain, since

$$X_i = |H_i|^2 S + |G_i|^2 N$$

holds, $\{a_i\}$ and $\{b_i\}$ are given by

$$a_i = |H_i|^2$$
$$b_i = |G_i|^2. \tag{5}$$

However unlike in the log-power-spectrum domain, $|H_i|$ and $|G_i|$ are independent, they can not uniquely related to the optimized λ_i.

2.3 Implementation

We have th following procedure to implement the technique. Log-power-spectrum is calculated through Mel-filter-bank analysis followed by log operation[8]. The spectrum of the speech captured by the close-talking microphone, X_0, is used as the speech at the source position *S*. All log-power-spectrum $\log X_i$ are normalized so that their means over an utterance become zero, i.e.,

$$\log X_i^{(d)} \approx \log X_i - \overline{\log X_i}.$$

Note that in this implementation, the minimisation of regression error is equivalent to minimising the MFCC distance between the approximated and the target spectra, due to the orthogonality of the discrete time cosine transform (DCT) matrix. Therefore, the MRLS has the same form as the maximum likelihood optimization of the filter-and-sum beamformer proposed in [5].

3. AUTOMATIC ADAPTATION OF MRLS

In the previous report[4], we found that changing regression weights adaptively to the driving conditions is effective in improving the recognition accuracy. In this section, we propose a method of discriminating in-car noise conditions, which is mainly affected by driving conditions, using *spatial distribution* of noise signals, and of controlling the regression weights for MRLS. The basic procedure of the proposed method is as follows. 1) Cluster the noise signals, i.e., short-time non-speech segments preceding utterances, into several groups. 2) For each noise group, train optimal regression weights for MRLS, using the speech segments. 3) For unknown input speech, find a corresponding noise group from background noise, i.e., the non-speech segments, and perform MRLS with the optimal weights for the noise cluster.

If there is a significant change in the sound source location, it greatly affects the relative intensity among distributed microphones. Therefore, in order to cluster the spatial noise distributions, we have developed a feature vector based on the relative intensity of the signals captured at the different positions to that of the nearest distant microphone, i.e.,

$$\mathbf{R} = [R_3(k), R_4(k), R_5(k), R_7(k)] \quad k = 4, 5, \cdots 24,$$

where $R_i(k) = X_i(k)/X_6(k)$ is the relative power at the k^{th} mel-filterbank (MFB) channel calculated from the i^{th} microphone signal. We do not use

	(1)normal	(2)music	(3)fan lo.	(4)fan hi.	(5)opn win.
cluster 1					
idle	**545**	10	0	0	232
city	**784**	69	130	8	100
highway	**895**	111	190	0	40
cluster 2					
idle	328	**873**	7	0	3
city	109	**827**	1	2	1
highway	3	**777**	5	2	0
cluster 3					
idle	24	15	**890**	**900**	28
city	0	2	**769**	**886**	5
highway	1	3	**695**	**898**	2
cluster 4					
idle	3	2	3	0	**637**
city	7	2	0	0	**794**
highway	1	9	2	0	**858**

Table 19-1. Distributions of the noise samples in the four clusters.

the lower frequency channel because the spectra of stationary car noise is concentrated in the lower frequency region. Thus, **R** is a vector with 84 elements. As shown in Figure 19-1, the 6^{th} microphone is the one nearest to the driver. Finally, the 84 elements are normalized so that their mean and variance across elements are 0 and 1.0, respectively. Prototypes of noise clusters are obtained by applying the k-means algorithm to the feature vectors extracted from the training set of noise signals.

An example of the clustering results are illustrated in Table 19-3, where we how many samples of each driving condition each noise class contains when four clusters of noise are learned. As seen from the table, clusters are naturally formed for 'normal', 'music playing', 'fan' and 'open window' situations, regardless of the driving speeds. From the results, it is expected that the relative power of the sound signals at different microphone positions can be a good cue for controlling MRLS weights.

4. IN-CAR SPEECH CORPUS FOR DISTRIBUTED MICROPHONE

The distributed microphone speech corpus is a part of the CIAIR (Center for Integrated Acoustic Information Research) in-car speech database collected at Nagoya University [7], which contains data from 800 speakers. They include isolated word utterances, phonetically balanced sentences and dialogues recorded while driving. The data collection is performed using a specially designed data collection vehicle that has multiple data acquisition capabilities of up to 16 channels of audio signals, three channels of video and other driving-related information, i.e., car position, vehicle speed, engine speed, brake and acceleration pedals and steering handle.

Five microphones are placed around the driver's seat, as shown in Figure 19-3, where the top and the side views of the driver's seat are depicted. Microphone positions are marked by the black dots. While microphones #3 and #4 are located on the dashboard; #5, #6 and #7 are attached to the ceiling. Microphone #6 is closest to the speaker. In addition to these distributed microphones, the driver wears a headset with a close-talking microphone (#1).

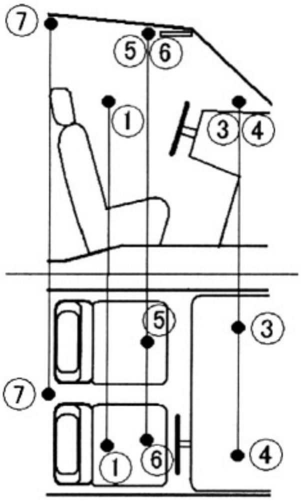


Figure 19-3. Microphone positions for data collection inside the vehicle: Side view (top) and top view (bottom).

In the majority of the corpus, the speaker is driving in the city traffic near Nagoya University. Considerable part of the corpus that we use in this study was collected under carefully controlled driving conditions, i.e., combinations of three car speeds (idle, driving in a city area and driving on an expressway) and five car conditions (fan on (hi/lo), CD player on, open window, and normal driving condition). For this part of the corpus, 50 isolated word utterances of 20 speakers were recorded under all combinations of driving speeds and vehicular conditions.

5. EXPERIMENTAL EVALUATIONS

5.1 Experimental Setup

Speech signals used in the experiments were digitized into 16 bits at the sampling frequency of 16 kHz. For the spectral analysis, 24-channel mel-filterbank analysis is performed by applying the triangular windows on the FFT spectrum of the 25-ms-long windowed speech. This basic analysis is realized through HTK standard MFB analysis [8]. The regression analysis is performed on the logarithm of MFB output. Since the power of the in-car noise signal is concentrated in the lower frequency region, the regression analysis is performed for the range of 250-8kHz, i.e., 4^{th} to 24^{th} spectral channels of the MFB. Then DCT is executed to convert the log-MFB feature vector into the MFCC vector for the speech recognition experiments.

Three different HMMs are trained:

close-talking HMM is trained using the close-talking microphone speech,

distant microphone HMM is trained using the speech at the nearest distant microphone, and

MRLS HMM is trained using MRLS results.

The regression weights optimized for each training sentence are used for generating the training data of MRLS HMM.

The structure of the three HMMs is fixed, i.e., three-state triphones based on 43 phonemes that share 1000 states; each state has 16-component mixture Gaussian distributions; and the feature vector is a 25 $(12 \text{ MFCC} + 12\ \Delta \text{ MFCC} + \Delta \text{ logpower})$-dimensional vector. The total number of training sentences is about 8,000. 2,000 of which were uttered while driving and 6,000 in an idling car.

5.2 Baseline Performance of MRLS

For the evaluation of the baseline performance of MRLS, five recognition experiments are performed:

CLS-TALK recognition of close-talking speech using close-talking HMM

MRLS SPKER recognition of MRLS output optimized for each speaker using MRLS HMM

MRLS DR recognition of MRLS output optimized for each driving condition using the MRLS HMM

MRLS ALL recognition of MRLS output optimized for all training data using MRLS HMM and

DIST recognition of nearest distant microphone speech by the distant microphone HMM.

The resulting recognition accuracies are listed in Table 19-4, and the average accuracies over fifteen driving conditions are shown in Figure 19-4. It is found that MRLS outperforms the nearest distant microphone result even in “MRLS ALL”, where a set of *universal* weights are used for all conditions. This result confirms the robustness of the MRLS to the change of the location of the noise sources, because the primary noise locations are different depending on driving conditions. It is also found that the improvement is greater when the performance of the distant microphone is lower.

5.3 MRLS Performance with Weight Adaptation

To evaluate the MRLS performance with weight adaptation, optimal regression weights for the four noise clusters of Section 3 are trained. Using a 200 ms non-speech segment preceding the utterance, the nearest prototype of the noise cluster is searched; then the utterance is recognized after MRLS with the regression weights optimized for the corresponding noise cluster using the same MRLS HMM. The results of the experiments are shown in Figure 19-5, where the performance of the MRLS using adaptive regression weights is as high as the results of using the optimally trained weights for each driving condition. Furthermore, the MRLS outperforms the MLLR adaptation (five-word supervised adaptation) applied to the close-talking speech [9]. Therefore, the effectiveness of the proposed method is confirmed.

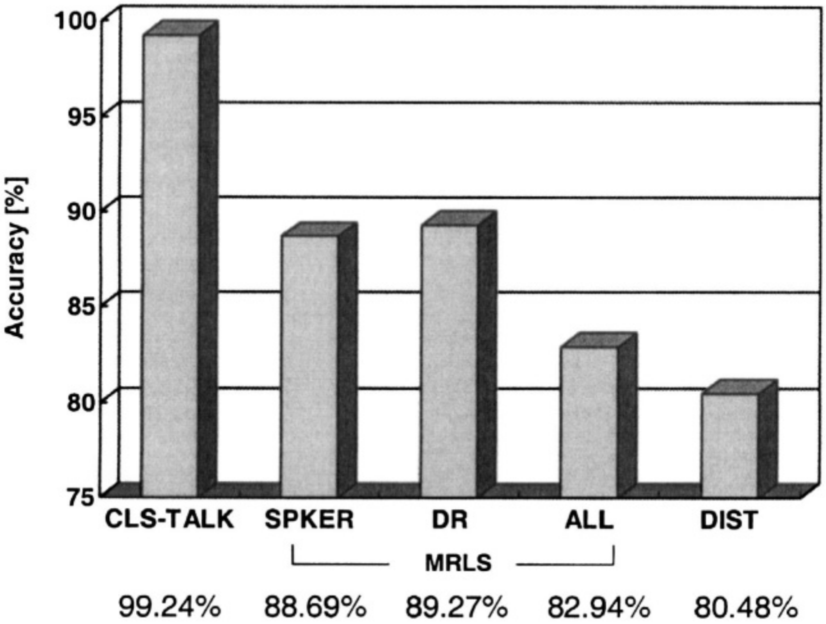


Figure 19-4. Recognition performance averaged over various driving conditions. Close-talking (CLS-TALK), MRLS with optimized weights for a speaker (SPKER), with optimized weights for each driving condition (DR), with optimized weights for all training data (ALL), MLLR and distant microphone (DIST), from left to right.

	(1)cls-talk	(2)mrls spker	(3)mrls dr	(4)mrls all	(5)dist.
		NORMAL			
idle	99.67	99.67	99.56	99.89	99.56
city	99.78	98.67	98.78	98.33	98.22
highway	99.56	96.56	97.00	92.56	92.44
		MUSIC PLAY			
idle	99.33	88.78	95.22	90.89	84.00
city	99.00	90.56	93.22	90.22	85.56
highway	99.78	91.56	92.89	88.89	86.89
		A.C. FAN ON LOW			
idle	98.56	98.11	98.33	97.00	95.00
city	99.89	97.89	97.44	95.00	95.11
highway	99.44	95.33	95.33	89.44	90.78
		A.C. FAN ON HIGH			
idle	98.89	75.22	76.22	59.44	53.89
city	98.55	78.79	79.58	65.51	61.38
highway	98.78	76.78	77.67	61.00	56.89
		OPEN WINDOW			
idle	99.56	95.67	95.44	92.56	88.33
city	98.89	86.22	85.56	77.11	75.78
highway	99.00	60.56	56.78	46.33	43.33

Table 19-2. MRLS accuracy results obtained under various driving conditions.

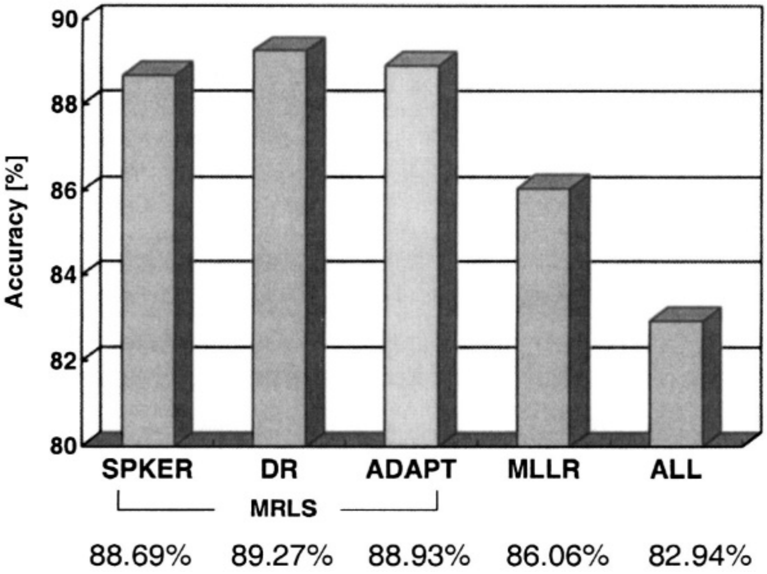


Figure 19-5. Recognition performance of MRLS with optimized weights for a speaker (SPKER), with optimized weights for a driving condition (DR), proposed weight adaptive method (ADAPT), with optimized weights for all training data (ALL), from left to right.

6. SUMMARY

In this paper, we described a multichannel method of noisy speech recognition that can adapt to various in-car noise conditions during driving. The method allows us to estimate the log spectrum of speech at a close-talking microphone based on the multiple regression of the log spectra (MRLS) of noisy signals captured by multiple distributed microphones. Through clustering of the spatial noise distributions under various driving conditions, the regression weights for MRLS are effectively adapted to the driving conditions. The experimental evaluation shows an error rate reduction of 43 % in isolated word recognition under various driving conditions.

Acknowledgement: This work has been supported by a Grant-in-Aid for COE Research (No. 11CE2005).

References

[1] Widrow, B. et al., "Adaptive Noise Cancelling: Principles and Applications", Proc. IEEE, Vol.63, No.12, (1975.12).

[2] Kaneda, Y. and Ohga, J., "Adaptive Microphone-Array System for Noise Reduction", IEEE Trans. Acoustics Speech and Signal Processing, 34 (6): 1391-1400, (1986).

[3] Yamada, T., Nakamura, S. and Shikano, K. "Distant-talking speech recognition based on a 3-D Viterbi search using a microphone array", IEEE Transactions on Speech and Audio Processing, Vol.10, No.2, pp.48-56, February 2002

[4] T.Shinde K. Takeda and F. Itakura, "Multiple regression of Log Spectra for in-car speech recognition", Proc. International Conference on Spoken Language Processing, Vol.II, pp.797-800, 2002 (ICSLP2002, Denver)

[5] Michael L. Seltzer, Bhiksha Raj, and Richard M. Stern, "Speech Recognizer-based microphone array processing for robust hands-free speech recognition", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol.I, pp.897-900, 2002 (ICASSP2002, Orlando)

[6] Shimizu, Y., Kajita, S., Takeda, K. and Itakura, F., "Speech Recognition Based on Space Diversity Using Distributed Multi-Microphone", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol.III, pp.1747-1750, (ICASSP2000, June, 2000, Istanbul).

[7] Kawaguchi, N., Takeda, K., et al., "Construction of Speech Corpus in Moving Car Environment", Proc. International Conference on Spoken Language Processing, pp. 1281-1284, 2000 (ICSLP2000, Beijing, China).

[8] Young, S. et al. "The HTK Book"

[9] C.J.Leggetter and P.C.Woodland, “Flexible speaker adaptation using maximum likelihood linear regression,” Proc. of the ARPA Spoken Language Technology Workshop, 1995, Barton Creek

This page intentionally left blank

INDEX