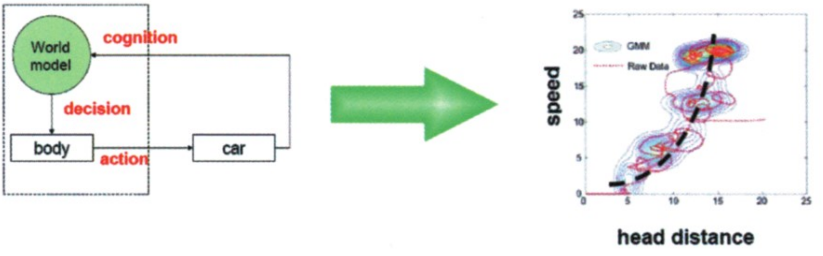
Advances for In-Vehicle and Mobile Systems

Challenges for International Standards



Edited by
Hüseyin Abut, John H.L. Hansen
and Kazuya Takeda

ADVANCES FOR IN-VEHICLE AND MOBILE SYSTEMS

Challenges for International Standards

ADVANCES FOR IN-VEHICLE AND MOBILE SYSTEMS

Challenges for International Standards

Edited by

Hüseyin Abut
San Diego State University, San Diego, California, USA and Sabancı University, Turkey
<abut@anadolu.sdsu.edu>

John H.L. Hansen
Center for Robust Speech Systems (CRSS)
Department of Electrical Engineering,
Erik Jonsson School of Engineering & Computer Science
University of Texas at Dallas, Richardson, TX, USA
<John.Hansen@utdallas.edu>

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

Hüseyin Abut
San Diego State University
San Diego, California, USA
and
Sabanci University
Istanbul, Turkey

John H.L. Hansen
Center for Robust Speech Systems (CRSS)
Department of Electrical Engineering
Erik Jonsson School of Engineering & Computer Science
University of Texas at Dallas
Richardson, TX, USA

Kazuya Takeda
Department of Media Science
Nagoya University
Nagoya, Japan

Advances for In-Vehicle and Mobile Systems: Challenges for International Standards

Library of Congress Control Number: 2004051229

ISBN-10 0-387-33503-X e-ISBN-10 0-387-45976-6
ISBN-13 978-0-387-33503-2 e-ISBN-13 978-0-387-45976-9

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

Foreword

The past, the present, and a glimpse of the future of the use of Digital Signal Processing (DSP) in vehicles are contained in the pages of this textbook. The papers within its covers present the results of an impressive array of research built on a solid base of 40 years of signal processing and automatic speech recognition. You will read papers that push parameters, tease out nuances, and point the reader into new directions as they address various aspects of the complicated man-machine interface that occurs in a moving vehicle requiring extensive cognitive demands. It can be a daunting interface, one that encompasses a challenging and changing acoustical environment with shifting situational wants and needs for drivers.

The past of DSP is clearly here. Prior research is often cited, and a knowledgeable reader will also see it in the starting points, assumptions, and techniques employed. The present of DSP is also certainly here. What began with few people, limited resources, and little public attention has mushroomed extensively in a relatively short period. The advent of such technologies as cellular telephony and accurate, interactive GPS navigational systems has made the public aware of the possibilities of DSP, and with public knowledge has come public demand. So, what had been the interest and passion of a few is now the pursuit of many. Public demand for voice-activated systems for in-vehicle environments for the sake of comfort and convenience is growing exponentially. The research presented in this book is intended to meet the public's demand in an effective, unobtrusive, and responsible way. Some of the topics covered include reduction, suppression, and control of noise in vehicles to enhance speech discrimination; speech recognition systems to facilitate speech control-oriented tasks; biometric recognition systems to identify individual drivers; and provide dialogue

management based on driver workload. Each research area discussed faces unique technical challenges in its development and implementation.

What's more, each research area also faces a special challenge because human beings are involved. As these papers show, the science has clearly advanced, but the capabilities of human users have not. For example, think about the complexity of developing a speech recognition system to enhance telematics use and how a driver might employ that system. Then think about the relatively simple, basic driving task. We know it is relatively simple by virtue of the sheer number of people who can successfully drive a vehicle. Millions, from teenagers to grandparents, drive. The availability of voice-activated telematics may well enhance the comfort, convenience, and, in some circumstances, the safety of drivers while they are carrying out this basic task. Still, we also have millions of accidents and tens of thousands of deaths each year, suggesting that the driving task is not always a simple one. When it is not, when the workload is high and time window is short, many drivers are ill-equipped to meet these changing and increased demands. What role does the voice-activated telematics system have at those times?

The answer to that question brings us to the future of DSP. Technologically, solutions are evolving rapidly, thanks to the bright men and women reporting on their research in this book. Future design and development will lead to a seamless implementation of hands-free interaction among digital devices in vehicles. In addition, systems will not only meet the wants and needs of drivers but also accommodate their frailties by factoring in the workload context. Safety demands require, and each researcher and system designer has to be ever mindful that, to be successful, these efforts must first do no harm.

So, read on, learn, be impressed, and see the future.

Bruce A. Magladry, Director
Office of Highway Safety
National Transportation Safety Board, USA

Contents

Contributing Authors

Abdul Wahab, Nanyang Technological University, Singapore

Markus Ablaßmeier, Munich University of Technology, Germany

Hüseyin Abut, San Diego State University, USA and Sabancı University, Turkey

Masato Akagi, Advanced Institute of Science and Technology, Japan

Ichiro Akahori, Denso Corporation, Japan

Murat Akbacak, University of Texas at Dallas, USA

Kathryn Arehart, University of Colorado at Boulder, USA

Jean-François Bonastre, Laboratoire Informatique Avignon, France

Paolo Bucciol, Politecnico di Torino, Italy

Luis Buera, University of Zaragoza. Spain

Francois Charpillet, Systems and Transportation Laboratory, France

Hideharu Daigo, Automobile Research Institute, Japan

Paul Dalsgaard, Aalborg University, Denmark

Juan Carlos De Martin, Politecnico di Torino, Italy

Aytül Erçil, Sabancı University, Turkey

Hakan Erdoğan, Sabancı University, Turkey

Franck Gechter, Systems and Transportation Laboratory, France

John H. L. Hansen, University of Texas at Dallas, USA

Nobuo Hataoka, Hitachi Advanced Research Laboratory, Japan

Keita Hayashi, Nagoya University, Japan

Yuki Irie, Nagoya University, Japan

Shunsuke Ishimitsu, University of Hyogo, Japan

Fumitada Itakura, Meijo University, Japan

Katsunobu Itou, Nagoya University, Japan

Shubha Kadambe, Signal and Image Processing, Office of Naval Research/University of Maryland, College Park, USA

Shingo Kato, Nagoya University, Japan

Nobuo Kawaguchi, Nagoya University, Japan

Kei Kikuiri, NTT DoCoMo Multimedia Laboratories, Japan

Géza Kiss, Budapest University of Technology and Economics, Hungary

Jeonwoo Ko, Carnegie Mellon University, USA

Abderrafiaa Koukam, Systems and Transportation Laboratory, France

Eduardo Lleida, University of Zaragoza. Spain

Christophe Lévy, Laboratoire Informatique Avignon, France

Junfeng Li, Advanced Institute of Science and Technology, Japan

Weifeng Li, Nagoya University, Japan

Georges Linarès, Laboratoire Informatique Avignon, France

Ana López, University of Zaragoza. Spain

Børge Lindberg, Aalborg University, Denmark

Toshiki Maeno, Nagoya University, Japan

Bruce A. Magladry, Office of Highway Safety, National Transportation Safety Board, USA

Enrico Masala, Politecnico di Torino, Italy

Enrique Masgrau, University of Zaragoza. Spain

Antonio Miguel, University of Zaragoza. Spain

Ralf Mathetat, Århus Technology Institute, Denmark

Shigeki Matsubara, Nagoya University, Japan

Teruko Mitamura, Carnegie Mellon University, USA

Chiyomi Miyajima, Nagoya University, Japan

Mitsunori Mizumachi, Kyushu Institute of Technology, Japan

Hiroya Murao, SANYO Electric Company, Japan

Fumihiko Murase, Carnegie Mellon University, USA

Nobuhiko Naka, NTT DoCoMo Multimedia Laboratories, Japan

Géza Németh, Budapest University of Technology and Economics, Hungary

Takuya Nishimoto, The University of Tokyo, Japan

Yoshihiro Nishiwaki, Nagoya University, Japan

Katsuyuki Niyada, Kyushu Institute of Technology, Japan

Pascal Nocera, Laboratoire Informatique Avignon, France

Sergio Nogueira, Systems and Transportation Laboratory, France

Eric Nyberg, Carnegie Mellon University, USA

Tomohiro Ohno, Nagoya University, Japan

Tomoyuki Ohya, NTT DoCoMo Multimedia Laboratories, Japan

Takahiro Ono, Nagoya University, Japan

Alfonso Ortega, University of Zaragoza. Spain

Koji Ozawa, Nagoya University, Japan

Pedro Ramos, University of Zaragoza. Spain

Gerhard Rigoll, Munich University of Technology, Germany

Yassine Ruichek, Systems and Transportation Laboratory, France

Hirohiko Sagawa, Hitachi Advanced Research Laboratory, Japan

Ana Salinas, University of Zaragoza. Spain

Antonio Servetti, Politecnico di Torino, Italy

Makoto Shioya, Hitachi Systems Development Laboratory, Japan

Juhei Takahashi, Automobile Research Institute, Japan

Kazuya Takeda, Nagoya University, Japan

Chin Keong Tan, Nanyang Technological University, Singapore

Zheng-Hua Tan, Aalborg University, Denmark

Masahiko Tateishi, Denso Corporation, Japan

Bálint Tóth, Budapest University of Technology and Economics, Hungary

Roberto Torrubia, University of Zaragoza. Spain

Luis Vicente, University of Zaragoza. Spain

Toshihiro Wakita, Toyota Central R&D Laboratories, Japan

Haitan Xu, Aalborg University, Denmark

Yukiko Yamaguchi, Nagoya University, Japan

Xianxian Zhang, University of Texas at Dallas, USA

Zhipeng Zhang, NTT DoCoMo Multimedia Laboratories, Japan

Introduction

In September 2005, the "Second Biennial Workshop on DSP (digital signal processing) for Mobile and Vehicular Systems" took place in Sesimbra, Portugal with 32 excellent papers present from all over the world, with a Keynote Address entitled "Information Overload and Driver Distraction: The Road to Disaster," delivered by Bruce A. Magladry, Director of the Office of Highway Safety, U.S. National Transportation Safety Board (NTSB), and a panel discussion with experts from academic, industry and federal agencies. This meeting represented a continuation from the first workshop in Nagoya, Japan, April 2003, and this book reflects the offspring of the 2005 workshop. After carefully reviewing all papers, 22 presentations from the workshop were selected and authors were asked to formulate extended book chapters of their original papers in order to provide a broad coverage of the fields in DSP for Mobile and Vehicular Systems, namely, Driver and Driving Environment Recognition, Telecommunication Applications, Noise Reduction and Dialog Systems for In-Vehicle systems. The chapters contained are therefore naturally categorized into four parts.

In the first part, a new and an emerging research field, Driver and Driving Environment Recognition is introduced and addressed with four complementary chapters. These chapters report on the application of signal processing technologies to characterizing human behavior while driving and encouraging further research efforts on the analysis of driver behavior.

The second part consists of three chapters dedicated to the most important application in Mobile and Vehicular Systems, (i.e., Telecommunication Applications). This part considers technologies for wireless communication systems, a distributed speech recognition paradigm, and embedded platforms.

The third and largest part of this book addresses a major challenge for Mobile and Vehicular Systems, namely Noise Reduction, which has ever-changing environmental noise that seriously degrades performance of these systems. Various research efforts that range from measuring and controlling the acoustic noise to an approach for audio-visual data fusion are discussed. ICA, Particle Filtering and other state of the art methodologies are also studied in this third part.

The topic of the last part of the book, Dialog Systems for In-Vehicle, is an essential issue for not only interface efficiency, but also for safety and comfort while driving. In this last part, six chapters discuss topics such as interaction between driving and dialogue, corpus, bimodal interface and dialogue strategies.

We hope this book will provide an up to date treatment of Mobile and Vehicular Systems, with new ideas for researchers and comprehensive set of references for engineers in related fields. We thank all those who participated in the 2005 workshop, we acknowledge support from the U.S. National Science Foundation and Nagoya University for their support in organizing the Biennial DSP for In-Vehicle and Mobile Systems in Sesimbra, Portugal, Sept. 2-3, 2005. We thank Dr. Pongtep Angkititrakul from CRSS-UTD for his assistance in book formatting/editing, and we wish to express our appreciation to Springer Publishing for ensuring a smooth and efficient publication process for this textbook.

The Editors, Hüseyin Abut, John H.L. Hansen, Kazuya Takeda

Chapter 1

EXPERIMENTS ON DECISION FUSION FOR DRIVER RECOGNITION

Hakan Erdoğan[1], Aytül Erçil[1] and Hüseyin Abut[1,2]
[1]Sabancı University, Istanbul, Turkey; [2]San Diego State University, San Diego, USA

Abstract: In this chapter, we study the individual as well as combined performance of various driving behavior signals on identifying the driver of a motor vehicle. We investigate a number of classifier fusion techniques to combine multiple channel decisions. We observe that some driving signals carry more biometric information than others. When we employ trainable combining methods, we can reduce identification error significantly using only driving behavior signals. Classifier combination methods seem to be very useful in multi-modal biometric identification in a car environment.

Key words: Biometric person identification, driver recognition, speaker recognition, face recognition, driving signals, driver behavior modeling

1. INTRODUCTION

Studies and technological advances in biometric person identification promise a world with no keys or passwords where smart devices or systems around us can identify us from our biological or behavioural traits. Biometric person identification would also be useful in a moving vehicle where most of us spend long hours every day. It is anticipated that the personalization of vehicles including driver identification for safety and comfort purposes [1] will be part of the picture in the near future.

In-vehicle person identification is a relatively new field of engineering science where only a few academic studies exist. Earlier, we have studied techniques to combine information from video, audio and driving signals to identify a driver of a vehicle using a 20-person subset of the Nagoya

University CIAIR database [1, 2]. In that study, we have used feature fusion of acceleration and brake pedal pressure signals to perform identification. Therefore, individual effects of acceleration and brake pedal pressure were not clear. However, in this chapter, we focus on individual identification performance of five different driving signals and their various combinations. We compare feature fusion versus decision fusion in this scenario as well.

This chapter is organized in the following way. In section 2, we describe the types of driving behaviour signals of the CIAIR database. We present our statistical GMM models for the driving signals in section 3. Fusion methods are explained in section 4. We present our experimental results in section 5 followed by the conclusions and future plans in the final section.

2. TYPES OF DRIVING SIGNALS

The Center for Integrated Acoustic Information Research (CIAIR) at Nagoya University has built a multi-modal corpus inside a vehicle, where each driver was required to carry out conversations with three different dialog systems while driving [2]. Data from 12 audio channels and 3 video channels have been recorded for over 800 drivers, both female and male. They have also collected five different "driving behavior signals" during these sessions. Driving behaviour data is collected from five analog channels, each sampled at 1.0 kHz with an unsigned 16-bit format.

- Brake pedal pressure in Kgforce/cm^2: 0-50 kgforce/cm^2 is mapped to 0 - 5.0V and linearly digitized in the range 0 to 32767.
- Accelerator pedal pressure in kgforce/cm^2.: 0-50 kgforce/cm^2 is mapped to 0 - 5.0V and linearly digitized in the range 0 to 32767.
- Engine speed in rpm: 0 - 8,000 rpm is mapped to 0 - 5.0V and linearly digitized in the range 0 - 32767.
- Vehicle speed in kmh: 0 - 120 kmh is mapped to 0 - 5.0V and linearly digitized in the range 0 - 32767.
- Steering wheel angle in -1800^o to +1800^o; i.e., five CW and five CCW revolutions is linearly digitized in the range -32769 to 32767.

In this chapter, we have utilized these signals in an attempt to identify drivers. To extract features from these signals, we perform smoothing and noise removal in time domain followed by decimation. We also extract dynamic features by computing the first difference of time-domain samples. Frequency-domain or cepstral features are normally used in speech/speaker recognition tasks. For driving signals, however, we deal with time-domain signals after a smoothing stage to reduce noise. Unlike speech, there is no

evidence of periodic (pitch) and frequency-related information in these driving signals. Frequency-domain processing could be useful to remove noise in these signals. However, noise removal could be performed in time-domain as well. Thus, we only use the time-domain signal directly in this work.

We use statistical modelling to model these driving signals and their first differences. We provide the details of our modelling approach in the following section.

3. MODELLING TIME-SERIES SIGNALS

As many time-series signals are slowly varying, it is natural to assume quasi-stationarity for modeling purposes. This naturally leads to Hidden Markov Model type dynamic generative models to model time-series data. In biometric identification from time-series data, the underlying state topology of the signal is usually unclear (except in the case of text-dependent speaker recognition) and single-state probabilistic models perform well. This is true given that we use a parametric continuous distribution function with multiple modes to cover variations in a time-series signal. Gaussian mixture models (GMM) are models that can approximate any smooth distribution, even if it has multiple modes. As long as we use enough number of mixtures in a GMM, we can obtain a good statistical model of the time-series data under consideration. GMM modeling for driving signals were first used in [3]. Following that work, we have also used GMM models in [1] for modeling driving behavior.

In GMM modeling, features are considered as independent identically distributed random vectors drawn from a GMM distribution:

$$\mathrm{f}(\mathbf{x} \mid S_i) = \sum_{k=1}^{K} \pi_k \, \mathrm{N}(\mathbf{x}, \boldsymbol{\mu}_\mathbf{k}, \boldsymbol{\Sigma}_\mathbf{k}). \tag{1}$$

where $\mathbf{x}$ represents the feature vector, π_k are mixture weights and $\mathrm{N}(\mathbf{x}, \boldsymbol{\mu}_\mathbf{k}, \boldsymbol{\Sigma}_\mathbf{k})$ are individual Gaussians for representing a particular subject under study, S_i. For computational purposes, $\boldsymbol{\Sigma}_\mathbf{k}$ are chosen to be diagonal matrices. GMMs have been used in text-independent speaker recognition applications with great success [4]. A popular way of using GMMs in speaker recognition is to train a large background speaker model, i.e., 1024 Gaussians, and adapt this model to each speaker using that particular

speaker's data. GMM training is performed using the well-known EM algorithm [5].

In this chapter, we train a GMM for each person's time-series data from scratch with eight (8) mixtures, which resulted in satisfactory performance in this application. During the testing phase, the per-frame log-likelihood value of observed data $\mathbf{x} = (\mathbf{x}_j)_{j=1}^{N}$ under the model of a particular person S_i can be computed as:

$$\begin{aligned} L_i(\mathbf{x}) &= \frac{1}{N}\sum_{j=1}^{N} \log \mathrm{f}(\mathbf{x}_j \,|\, S_i) \\ &= \frac{1}{N}\sum_{j=1}^{N}\left(\log \sum_{k=1}^{K} \pi_k \, \mathrm{N}(\mathbf{x}_j, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right). \end{aligned} \tag{2}$$

As is customary in speech recognition research, we also train a background model, one more GMM, with twice the number of mixtures. Background GMM is required for normalization in likelihood ratio testing for biometric *verification*. The log-likelihood of the observed data under the background model, $L_g(\mathbf{x})$ can also be computed in a similar way.

In the *verification* task, the Bayesian decision amounts to the comparison of the log-likelihood-ratio, $L_i(\mathbf{x})$ -$L_g(\mathbf{x})$ to a pre-determined threshold. For different thresholds, we trace the receiver operating characteristics (ROC) curve which plots false-accept rate versus false-reject rate.

For *identification* problem, however, we need to obtain posterior probabilities of identities given test data and choose the largest one as the identity of the test segment. The posterior probabilities can be found from:

$$p(S_i \mid \mathbf{x}) = \frac{e^{L_i(\mathbf{x})}}{\sum_{j=1}^{M} e^{L_j(\mathbf{x})}}, \tag{3}$$

where we assume equal priors for each class. This set of probabilities is also called ***scores***. We discuss how to combine these scores from different modalities in the following section.

4. FUSION METHODS

Combining multiple classifiers is a new applied research area that has attracted great interest [6] during the past few years. Classifier combination methods can be divided into two categories: fixed and trained.

Fixed methods have simple fixed rules to combine information from a set of classifiers. On the other hand, trainable combination methods have some free parameters that can be trained on a separate part of training data (validation or held-out data). It is not difficult to see that trainable combiners are classifiers themselves, which classify in the score space rather than the original feature space.

Given test data $\mathbf{x}$[1], let $S(i,j)$ represent the score of person i in modality j. Our goal is to obtain a single score $S(i)$ for person i using a combination method. We identify various classifier combination methods below:

4.1 Fixed combiners:

In a number of studies we have observed the use of simple fixed rules to combine scores from different classifiers (modalities). Fixed rules are suboptimal since classifiers are not differentiated among each other. Thus, fixed rules do not take into account the variability of reliabilities of different classifiers.

Some fixed rules of classifier combination are listed below:

- Max Rule: $S(i)=\max_j S(i,j)$
- Min Rule: $S(i)=\min_j S(i,j)$
- Mean (sum) Rule: $S(i)=\text{sum}_j S(i,j)$
- Product Rule: $S(i)=\text{prod}_j S(i,j)$
- Median Rule: $S(i)=\text{median}_j S(i,j)$

4.2 Trainable combiners:

In trainable classifier combination framework, we form a vector of all scores computed using all the classifiers available. The entries of the vector is given as the elements of the following set: $S=\{S(i,j): i=1..N_p, j=1..N_c\}$ where N_p and N_c denote the number of people (classes) and classifiers respectively.

This score vector is used as a new feature vector for classification. Thus, we can use any classification method as a second classifier. It is worth

[1] We drop $\mathbf{x}$ from our notation for brevity.

noting that the second classifier should be trained using a different (held-out or validation) data set from the original training data to avoid overtraining.

We have used the following types of combining classifiers in this work:

- Nearest mean combiner (NMC): A simple linear combiner that chooses the nearest class mean as the classifier output.
- Fisher combiner (Fisher): A linear classifier that minimizes the least squares error in mapping features to class labels in a one-vs-all fashion.
- Linear discriminant combiner (LDC): Another linear classifier that models each class by a Gaussian that shares the same covariance matrix with other classes.
- Naïve Bayes combiner (NB): It assumes that the class-conditional probabilities of the feature vector coordinates are statistically independent. Each coordinate is modeled with a nonparametric binning distribution model with 10 bins.
- Parzen combiner: Parzen density based combiner.

5. EXPERIMENTS AND RESULTS

We have performed experiments on decision fusion for driver recognition using a subset of the extensive CIAIR database from Nagoya University, which consisted of a 20 person subset of the database that we have also used in our earlier work [1]. In this current study, however, we extract features from each and every driving signal and evaluate their performance individually as well as after combination.

50 image frames, 50 seconds of non-silence audio, and approximately 600 seconds of driving signals were utilized from each driver. We have divided features into 20 equal length segments for each driver and modality and have labeled the segments from one to 20. Then we have formed the multimodal test-sets where it was assumed that each modality segment was associated with segments that have the same number in other modalities. Smoothed and sub-sampled driving signals and their first derivatives were used as features for modeling driving behavior of the drivers. Thus, each driving signal was composed of two-dimensional feature vectors.

The training procedure was a leave-one-out type, where for each single testing segment, seventeen parts were used for training and two parts were held-out for validation to optimize normalization parameters and fusion weights. This gave us 20 tests for each person (each time the training data is different although not independent), leading to 400 (20x20) genuine tests in total. GMMs were driven with eight, one, and eight mixture components for speech, face, and driving signals, respectively. Background GMM models were trained for each modality as well [6].

We have performed closed set identification for this the dataset. *prtools* [7] software library were used in evaluating the results and combining the classifiers. In Table 1-1, individual performance results for each (possibly feature combined) modality are tabulated.

Table 1-1. Individual performance results for different modalities.

Modality	Percent Error (%)
Acceleration (A)	42.5
Brake (B)	31.7
Engine Speed (E)	84.2
Vehicle Speed (V)	81
Steering wheel angle (W)	88.7
A+B+E+V+W	31.2
A+B[2]	10.2
A+B+W	16.5
Speech (S)	2
Face (F)	11

In this table, + sign denotes feature fusion, that is, A+B means that acceleration and brake features are concatenated and a larger feature vector of dimension 4 is obtained. The results clearly show that every single driving signal is not individually appropriate for biometric identification. However, feature fusion of acceleration and brake signals (A+B) yields a respectable 10.2% error rate. This was also observed in [1].

In Table 1-2, we present results from five different decision fusion experiments using fixed rules for various combination of modalities. In this table the comma (,) sign indicates decision fusion, that is, classifier posterior probabilities are combined.

Table 1-2. Error rates (%) for fixed combination rules in combining different modalities.

Modalities	Max	Min	Median	Mean	Product
A,B	28.2	14.5	14	14	11.2
A,B,E,V,W	43	31	41.5	22.7	23.5
A,B,W	38	22.5	30.7	18.7	16.2
A,B,F,S	9	3.2	0	0	1

[2] The results for A+B, F and S features were found in [1]. For A+B driving features, we re-estimated the GMMs. Due to random initialization, the results are slightly different than the ones reported in [1].

The fixed-combination rules are generally suboptimal since they do not consider relative reliability of individual modalities. We observe that among the group of fixed combiners, product and mean rules perform the best in general. Since acceleration and brake tend to be reliable amongst driving signals, it was possible to achieve better results by just using these two rather than all driving signals. When we have added face and speech modalities, we could achieve close to 0% error even with these suboptimal fixed-combination rules.

In Table 1-3, trainable combiner results are presented. The trainable combiners are trained using validation data that was set aside from training and testing data in the cross-validation procedure described above.

Table 1-3. Error rates (%) for trainable combination methods.

	NMC	Fisher	LDC	NB	Parzen
A,B	12.7	11.7	10.7	6.5	0.2
A,B,E,V,W	10.5	5.2	3	5	0
A,B,W	10	8.2	7	6.7	2
A,B,F,S	0	0	0.2	0	0

In trainable combiners, we have generally achieved lower error rates, which can be attributed to validation data training. In this study, it appears that the validation and test data are very similar and over-training combiners such as Parzen density based combiner work is clearly the best. It is well known fact in pattern recognition community that Parzen classifier tend to overfit to training data and does not easily generalize. Nevertheless, in our experiments, it is clearly the most promising choice among the combiners studied. Linear combiners, such as the LDC and Fisher types, also yield respectable performance especially when the numbers of input classifiers in the combination are large.

6. CONCLUSIONS AND FUTURE WORK

In this chapter, we have studied the performance of various combination methods for driver identification using driving behavior signals collected in a real-world scenario. The results show that individual driving signals are not largely indicative of the person by themselves. However, when we fuse decisions from GMM classifiers of driving signals using trainable combiners, we can achieve significantly low error rates in identifying the driver of the vehicle.

In the future, we plan to test these models in a larger subset of the CIAIR database.

ACKNOWLEDGEMENT

The authors would like to thank Professors Kazuya Takeda and Nobuo Kawaguchi of Nagoya University and Professor Fumitada Itakura of Meijo University for introducing to the problem and providing the CIAIR database.

REFERENCES

[1] H. Erdogan, A. Ercil, H.K. Ekenel, S.Y. Bilgin, I. Eden, M. Kirisci, H. Abut, "Multi-modal person recognition for vehicular applications," N.C. Oza et al. (Eds.): MCS 2005, LNCS 3541, pp. 366 – 375, Monterey CA, Jun. 2005.

[2] N. Kawaguchi, S. Matsubara, I. Kishida, Y. Irie, H. Murao, Y. Yamaguchi, K. Takeda and F. Itakura, "Construction and Analysis of the Multi-layered In-car Spoken Dialogue Corpus," Chapter 1 in *DSP in Vehicular and Mobile Systems,* H. Abut, J. H.L. Hansen, and K. Takeda (Editors), Springer, New York, NY, 2005.

[3] K. Igarashi, C. Miyajima, K. Itou, K. Takeda, H. Abut and F. Itakura, "Biometric Identification Using Driving Behavior," Proceedings IEEE ICME 2004, June 27-30, 2004, Taipei, Taiwan.

[4] D.A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," Speech Communications, 17, 91-108, 1995.

[5] A Dempster, N Laird, M Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", J. Royal Statistical Soc., 39, 1, 1978.

[6] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226-239, 1998.

[7] R.P.W. Duin, P. Juszczak, D. de Ridder, P. Paclik, E. Pekalska, and D.M.J. Tax, PRTools, a Matlab toolbox for pattern recognition, http://www.prtools.org, 2004.

Chapter 2

DRIVER RECOGNITION SYSTEM USING FNN AND STATISTICAL METHODS

Abdul Wahab[1], Tan Chin Keong[1], Hüseyin Abut[2], and Kazuya Takeda[3]
[1]School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore (639798); [2]ECE Department, San Diego State University, San Diego, CA 9218, USA and Sabanci University, Istanbul, Turkey; [3]School of Information Science, Nagoya University, Japan.

Abstract: Advancements in biometrics-based authentication have led to its increasing prominence and are being incorporated into everyday tasks. Existing vehicle security systems rely currently on electronic alarm or smart card systems. A biometric driver recognition system utilizing driving behavior signals can be incorporated into existing vehicle security system to form a multimodal identification system and offer a higher degree of protection. The system can be subsequently integrated into intelligent vehicle systems where it can be used for detection of any abnormal driver behavior with the purposes of improved safety or comfort level. In this chapter, we present features extracted using Gaussian Mixture Models (GMM) from accelerator and brake pedal pressure signals, which are then employed as input to the driver recognition module. A novel Evolving Fuzzy Neural Network (EFuNN) was used to illustrate the validity of the proposed system. Results obtained from the experiments are compared with those of statistical methods. They show potential of the proposed recognition system to be used in real-time scenarios. A high identification rate and the low verification error rate were indicated considerable difference in the way different drivers apply pressure to the pedals.

Key words: driving profile, behavioral modeling, verification and identification, soft computing, accelerator and brake pressure, dynamic driver profiling

1. INTRODUCTION

Biometric Identification is a broad category of technologies that performs automatic recognition of an individual based on the individual's physiological or behavioral characteristics. Physiological characteristics are

relatively stable physical features, such as fingerprint, iris, facial features or hand geometry[1-4] while behavioral characteristics are affected, usually in a complex fashion, by the individual's mental status and they include voiceprint, hand-written signature or keystroke dynamics[1-3]. The first class of biometrics, in particular fingerprint, has been widely used in forensics and now being evaluated in banking transactions. The second class of biometrics is gaining prominence in recent years with speaker/face/gait recognition garnering the most attention[4].

In a recent work, driving characteristics, in particular, the amount of pressure a driver applies on the accelerator pedal and/or the brake pedal have been utilized in personal identification[5]. Encouraging experimental results indicate that there is uniqueness in driving behavior among individuals. The utilization of driving behavioral signals can blend nicely with the existing vehicle security systems and offer a higher degree of multi-level protection. Additionally, the recognition system can be integrated into intelligent vehicle systems with purpose of achieving safer driving. For example, upon recognition of the driver by the system, a profile of the driver can be loaded from the system associative memory. Any deviation of the driver behavior from its norm can then be identified and necessary actions could be taken accordingly.

Artificial Neural Networks has emerged as a powerful and practical computing tool over recent years, particularly in the field of pattern recognition/classification[6]. Two limitations associated with most artificial neural networks are their long training process and finding an optimal boundary when handling real-life data due to the ambiguous/ever changing nature of such data. Fuzzy logic was introduced as an approach to handling vagueness and uncertainty[7]. Fuzzy neural hybrid systems combine the two concepts by applying learning techniques of neural networks for fuzzy models parameter identification. These systems offer strong generalization ability and fast learning capability for large amount of data. Even though still not widely explored, fuzzy neural systems like the Evolving Fuzzy Neural Network (EFuNN) have been applied in several recognition studies with high degree of accuracy[8,9]. In this study, the performance of an EFuNN will be compared to Gaussian Mixture Statistical Scheme (GMSS) on driver recognition tasks. The prior work by Igarashi and others[5] will also be implemented for comparison.

1.1 Resources

Driving data utilized in this research are subset of the In-car Signal Corpus collected by the Center for Integrated Acoustic Information Research (CIAIR), Nagoya University, Japan[10]. The In-car Signal Corpus is one of

several databases hosted by CIAIR. This database contains multi-dimensional data collected in a vehicle under both driving and idling conditions. The purpose of setting up the database was to deal primarily with the following two issues: noise robustness of speech and continual change of the vehicular environment. To date, the number of subjects involved in the data collection is more than 800 (men and women) with a total recording time of over 600 hours. The multimedia data consists of speech, image, control (driving) and location signals, all synchronized with the speech channels. For this research, only the driving signals (accelerator pedal pressure and brake pedal pressure) were utilized.

Modeling and studies of driving behaviors began as early as in the 1950s. Many of the studies have been conducted with the objectives of increasing traffic safety or improving the performance of intelligent vehicle systems[11,12,13]. However, the utilization of driving behavior for personal identification is still not widely explored.

2. DATA ANALYSIS AND FEATURE EXTRACTION

Vehicle control signals from the In-car Signal Corpus consist of accelerator pedal pressure, brake pedal pressure, steering angle, engine speed and vehicle speed. The first three are more driver-dependent traits while the latter two are more vehicle-dependent attributes. In this research, the focus was placed only on the driver-dependent traits and more specifically, the accelerator pedal pressure and brake pedal pressure signals since it was noted that there is considerable differences among drivers in the way they apply pressure to the pedals. The vehicle control signals were collected through analog channels, each sampled at 1.0 kHz with a 16-bit resolution. The pedal pressure sensors can detect pressures ranging from 0.0-30.0 kgforce/cm^2. This range is mapped to 0 – 5.0 V and linearly digitized in the range 0 to 32767.

Segments known as stop&go regions were extracted from the original driving signals used in the experiments conducted by Igarashi and others[5]. These segments are also available in the In-car signal corpus. A stop&go region is defined as the period the vehicle starts moving it comes to a complete halt. The motivation for using just the stop&go regions instead of the entire signals is instinctive since little or no information pertaining to driving behaviors is present when the vehicle is not in motion. Figures. 2.1-2.3 show the associated vehicle speed, accelerator pedal pressure and brake pedal pressure signals of a stop&go region. At the start, the vehicle speed remains at 0 for a brief amount of time. This indicates that the vehicle is in a halted state.

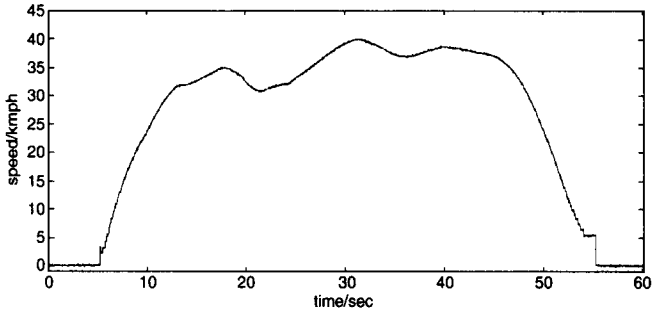


Figure 2-1. Signal trajectory of a stop&go region for the vehicle speed.

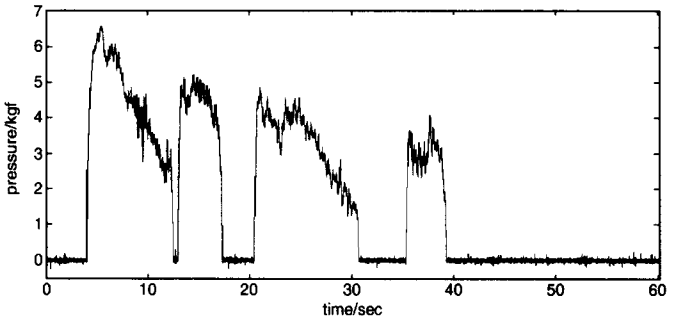


Figure 2-2. Signal trajectory of a stop&go region for the accelerator pedal pressure.

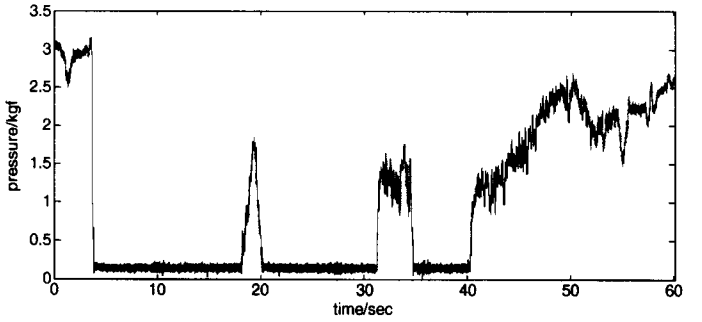


Figure 2-3. Signal trajectory of a stop&go region for the brake pedal pressure.

It can be seen from the brake pedal pressure signal plots that the driver was applying pressure on the brake pedal during the period of time when the vehicle is stationary. Shortly after, the brake pedal pressure goes to zero and there is a sharp transition in the accelerator pedal pressure signal. The vehicle speed then increased quite constantly for about 15 seconds before a slight drop in the vehicle speed. This portion of the stop&go region is sometimes referred to as the initial-acceleration. The vehicle then maintains at an average speed of about 35 kmh for approximately 30 seconds. This region during which the vehicle travels at a constant speed can be referred to as the steady state in which there is no significant variation in the vehicle speed. Following that, the vehicle speed starts to decrease gradually until the vehicle comes to a complete halt indicated by the vehicle speed signal. This region can be referred to as the deceleration or stopping region during which no pressure is applied to the accelerator pedal. As expected, at any given moment, the driver can apply pressure on only one of the pedals.

In studies, often the focus is not placed only on the static data but not on the dynamics of the data as well. Dynamics of pedal pressure can be defined as the rate of change in pressure applied on the pedal by the driver. Intuitively, this offers additional information on top of the static signals. In the research conducted by Igarashi et al, it was found that dynamics improve the performance of driver identification compared to when only the static signals were being used. Figure 2-4 shows an accelerator pedal pressure signal and its dynamics respectively. The dynamics signal is a function of time with the pressure/s^2 as the y-axis. The value at any point represents the rate of change in pedal pressure. For example, a sharp positive-going transition (increase) in the accelerator pedal pressure is translated to a high positive rate of change value in the dynamics; a sharp negative-going transition (decrease) in the pedal pressure is translated to a high negative rate of change value in the dynamics.

2.1 Feature Extraction

Reduction of data size is a critical step in the neural network approach to pattern recognition tasks. Pre-processing can often greatly improve the performance of a pattern recognition system. If a prior knowledge about the data is present, the performance can often be improved considerably by a selection of relevant features that can best characterize the data. In general, to obtain an appropriate model of the data and achieve faster learning, irrelevant information must be eliminated from the network training data.

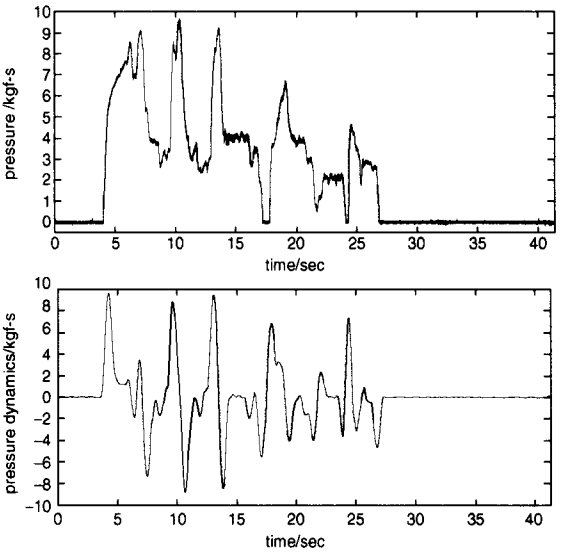


Figure 2-4. Accelerator Pedal Pressure Signal (top) and its Dynamics (bottom).

2.2 Gaussian Mixture Models

Gaussian Mixture Model is a semi-parametric approach to density estimation[6]. Besides offering powerful techniques for density estimation, Gaussian mixture models provide important applications in the context of neural networks, in techniques for conditional density estimation, soft weight sharing and in the mixture-of-experts model. Gaussian mixtures are also well known for their ability to form smooth approximations to arbitrarily shaped densities. The use of Gaussian mixture models for modeling driver identity is motivated from the observed behavior that there is a general tendency for the driver to exert certain amounts of pressure on the pedals more frequently than others and in some distributions that can be represented by Gaussian components.

3. EXPERIMENTAL SETUP

The driver recognition task was compared on different implementations of the system using GMSS and EFuNN. The training and testing methodology for the neural network-based implementation is first discussed. Features were extracted from the driving data (stop&go regions) of 30 drivers. For each driver, each set of features can be further classified into four sets corresponding to the signal under study, namely, the accelerator pedal pressure, brake pedal pressure, dynamics of accelerator pedal pressure

and dynamics of brake pedal pressure. Each driver can be modeled by a single or up to four networks corresponding to the different signal types.

Generally, two types of data files were prepared as the source for the networks: training data set and the test data. Identification is performed by presenting the testing data file(s) to the driver recognition system which is then presented to all the corresponding network(s) of all drivers. The networks' outputs are linearly combined for each driver and the driver with the highest combined network output is identified as the driver. For verification, the testing data file is fed to the asserted driver network and a linearly combined output of the network is compared with a decision threshold. If the output satisfies the pre-defined threshold level, the identity claim is verified otherwise the claim will be rejected.

Each driver is modeled by up to four sets of GMM parameters. Each set of GMM parameters is computed for a single vector formed by appending the stop&go regions designated for training. In general, there would be a total of ten driver templates corresponding to forty sets of GMM parameters for each driver recognition system. For identification, the input signal(s) are presented to the driver recognition system where the likelihood is measured for each driver template and the driver template that gives the maximum likelihood is identified as the driver. For verification, the input signal is presented to the driver template for which the claim is asserted and the likelihood is computed. If the likelihood value satisfies a pre-defined threshold level, the identity claim is verified otherwise the claim will be rejected.

3.1 Validation Method

Experiments were conducted on two groups of drivers where each group consisted of 10 different drivers and the average number of input patterns for each driver is 16. The *N-Leave-One-Out* validation method is employed in the experiments. Given N cases (stop&go regions) for each driver numbered from 1 to N, the validation is performed as follows:

1. The n^{th} case for each driver is omitted from the training process.
2. The omitted cases are used in the testing process.
3. Steps 1 and 2 are repeated for each case of the data set.

3.2 Driver Identification Performance

In the first set of experiments, the EFuNN-based driver recognition system was trained and tested using the GMM-based features. The performances of these implementations were measured against GMSS. The

identification results for two groups of drivers are presented below in Tables 2-1 and 2-2.

Table 2-1. Group I Identification Results based on GMM Features using both the accelerator and Brake pedal pressure.

Signals	Accelerator + Brake Pedal Pressure (Static & Dynamic)			
System Type	GMSS		EFuNN	
Driver	Accuracy [%]	Test time/s	Accuracy [%]	Test time/s
1	93.75	2.37	81.25	0.78
2	100	1.59	93.75	0.94
3	100	2.86	100	0.78
4	100	2.47	81.25	0.93
5	87.5	2.30	93.75	0.79
6	93.75	2.58	93.75	0.93
7	93.75	2.64	100	0.78
8	93.75	2.25	75	0.94
9	100	2.53	87.5	0.94
10	87.5	1.81	81.25	0.78
Average	95.0	2.34	88.75	0.86

Signals	Accelerator + Brake Pedal Pressure (Static)			
System Type	GMSS		EFuNN	
Driver	Accuracy [%]	Test time/s	Accuracy [%]	Test time/s
1	81.25	1.37	81.25	0.47
2	81.25	0.88	68.75	0.47
3	81.25	1.54	75	0.46
4	87.5	1.43	81.25	0.47
5	75	1.76	81.25	0.47
6	93.75	1.60	68.75	0.31
7	87.5	1.49	81.25	0.47
8	87.5	1.32	56.25	0.47
9	81.25	1.48	81.25	0.47
10	87.5	0.93	75	0.47
Average	84.38	1.38	75.0	0.45

It was observed from these tables that the fuzzy neural systems performed comparatively well against GMSS in terms of identification rate. The driver identification performance is also consistent among different tests. Among these two groups of drivers, the highest accuracy was obtained when the combination of all signals was used. It can be seen that the average identification rate obtained from using a version of Hui's ANFIS system[9] is very close to the rate obtained for GMSS. From the driver identification tests

several observations were made. The accuracy of the GMM-based systems is good and fairly consistent between the GMMSS and the EFuNN. It may be reasonable to infer from these results that the pressure distribution information can better characterize driving behavior. Additionally, driving behavior modeling based on the pressure distribution is a more natural and intuitive method.

Table 2-2. Group 2 Identification Results based on GMM Features using both the accelerator and Brake pedal pressure.

Signals	Accelerator + Brake Pedal Pressure (Static)			
System Type	GMSS		EFuNN	
Driver	Accuracy [%]	Test time/s	Accuracy [%]	Test time/s
1	87.5	1.48	93.75	0.63
2	93.75	1.04	100	0.62
3	62.5	1.10	87.5	0.63
4	81.25	2.47	87.5	0.62
5	68.75	1.54	68.75	0.63
6	87.5	2.03	81.25	0.62
7	87.5	2.14	81.25	0.78
8	81.25	1.26	81.25	0.63
9	75	1.16	68.75	0.62
10	81.25	1.32	68.75	0.47
Average	80.63	1.55	81.88	0.63

Signals	Accelerator + Brake Pedal Pressure (Static & Dynamic)			
System Type	GMSS		EFuNN	
Driver	Accuracy [%]	Test time/s	Accuracy [%]	Test time/s
1	100	2.30	100	1.56
2	100	2.52	100	0.94
3	75	1.75	100	0.78
4	93.75	4.29	87.5	1.09
5	81.25	2.64	68.75	0.94
6	100	3.41	87.5	0.94
7	93.75	3.74	87.5	0.94
8	93.75	2.09	93.75	0.78
9	100	1.97	93.75	0.93
10	87.5	2.20	68.75	0.94
Average	92.5	2.69	88.75	0.98

In terms of processing time, the training for EFuNN takes less than 20s on the worst case and testing time is only a fraction of a sec. The average testing time for GMSS was much longer compared to EFuNN systems. The testing times in all the different implementations were generally low, therefore indicating that the identification task can be performed in a relatively short amount of time. It can also be noted that the training time of the ANFIS systems were significantly smaller in GMM-based systems.

For both groups of drivers, the best performance was obtained when a combination of all of the driving data was used. The same phenomenon was observed in the work by Igarashi et al[5] indicating that the combination of these signals can characterize the driving behavior to a higher degree than other combinations of the signals. From these results, a driver identification system with high accuracy and fast testing time can be implemented using EFuNN with the combination of static and dynamic accelerator and brake pedal pressure signals.

3.3 Driver Verification Performance

In the second phase of testing, a further evaluation on the performance of the driver recognition system using these three configurations was carried out. The driver verification performance in terms of the equal error rate was measured for these two groups of drivers.

Table 2-3. Verification results of group 1 driver using the GMM features from the Accelerator + Brake Pedal Pressure (Static & Dynamic).

System Type	GMSS		EFuNN	
Driver	EER [%]	Test time(s)	EER [%]	Test time(s)
1	2.5	0.24	2.5	0.078
2	0	0.16	0	0.094
3	0	0.29	2.5	0.078
4	0	0.25	2.5	0.093
5	7.5	0.23	0	0.079
6	2.5	0.26	0	0.093
7	2.5	0.26	0	0.078
8	3.125	0.23	2.5	0.094
9	0	0.25	10	0.094
10	7.5	0.18	2.5	0.078
Average	2.5625	0.23	3.25	0.086

Table 2-4. Verification results of group 2 driver using the GMM features from the Accelerator + Brake Pedal Pressure (Static & Dynamic).

System Type	GMSS		EFuNN	
Driver	EER [%]	Test time(s)	EER [%]	Test time(s)
1	0	0.23	0	0.156
2	0	0.25	0	0.094
3	12.5	0.18	0	0.078
4	2.5	0.43	22.5	0.109
5	7.5	0.26	7.5	0.094
6	0	0.34	7.5	0.094
7	2.5	0.37	12.5	0.094
8	2.5	0.21	2.5	0.078
9	0	0.19	3.125	0.093
10	12.5	0.22	5.0	0.094
Average	4.0	0.27	6.0625	0.098

The performances of the three implementations are comparatively respectable in the verification task and thus indicate the driver recognition system's ability to deter most unauthorized access. Despite a reasonably good performance, this may not be sufficient especially for strict access control or security systems since any false acceptance will result in serious consequences. Despite the ability of the GMM-based features to model driving behavior to a high degree of accuracy, moderate verification results suggest that there is slight similarity in driving behaviors among drivers in terms of the way they apply pressure to the brake and accelerator pedals which requires more extensive investigation and research.

4. CONCLUSION AND RECOMMENDATIONS

In this research, statistical, artificial neural network, and fuzzy neural network techniques were implemented and compared in the framework of driver recognition. Gaussian Mixture Models was proposed and implemented. Features were extracted from the accelerator and brake pedal pressure signals of 30 drivers. These features were then used as input to a class of fuzzy neural network-based driver recognition systems, namely EFuNN. This system was compared against a well known statistical method, GMSS.

Extensive testing was carried out using Matlab and several observations were made. The use of the mean pressures (applied on the accelerator and brake pedals) obtained using the GMM-based extraction process as inputs to

the fuzzy neural-network based driver recognition systems was found to achieve a high identification rate and low verification equal error rate. These experimental results show that the use of the means solely out of the entire set of GMM parameters is adequate to efficiently characterize driving behavior. The combination of the accelerator pedal and brake pedal pressures and their dynamics was also found to give the best performance among driver combinations of the signals. Fuzzy neural systems, EFuNN performed comparatively well against GMSS. EFuNN offer fast testing time.

The notion of utilizing driving behaviors in biometric identification may initially appear to be a bit far-fetched but it has been shown to be realizable. This biometric method will offer not only an added level of protection for vehicles but also a natural and secured identification of drivers. The system can be subsequently integrated into intelligent vehicle systems where it can be used for detection of any abnormal driver behavior for the purpose of achieving safer driving experience. The field of driver recognition is a relatively new field of study which requires more research and investigations. Further exploration is required to refine and optimize the current system implementation.

REFERENCES

[1] S. Barua, "Authentication of cellular users through voice verification", Systems, Man, and Cybernetics, 2000 IEEE International Conference, Volume: 1, 8-11 Oct. 2000, Pages: 420 - 425

[2] I. Pottier and G. Burel, "Identification and authentication of handwritten signatures with a connectionist approach", *Neural Networks, IEEE World* Congress on Computational Intelligence, 1994 IEEE International Conference, Volume: 5, 27 June-2 July 1994, Pages: 2948 – 2951

[3] M.S. Obaidat B. Sadoun, "Verification of computer users using keystroke dynamics", Systems, Man and Cybernetics, Part B, IEEE Transactions, Volume: 27, Issue: 2, April 1997, Pages: 261 - 269

[4] D. Reynolds, "An overview of automatic speaker recognition technology", Acoustics, Speech, and Signal Processing, Proceedings ICASSP, IEEE International Conference, Volume: 4, 13-17 May 2002, Pages: 4072 – 4075

[5] K. Igarashi, K. Takeda, F. Itakura and H. Abut, "Biometric Identification Using Driving Behavioral Signals", Chapter 17 in *DSP for In-Vehicle and Mobile Systems*, Springer Science, Publishers, New York, 2005.

[6] C. Bishop, Neural Networks for Pattern Recognition, Oxford: Clarendon Press, 1995

[7] L. A. Zadeh, "Fuzzy Logic", Computer, Vol. 1, No. 4, Pages: 83-93, 1988

[8] L. Zhang, "Associative Memory and Fuzzy Neural Network for Speaker Recognition", Unpublished Honor Year Project Report, School of Computer Engineering, Nanyang Technological University, Singapore 2004

[9] H. Hui, J.H. Li, F.J. Song, and J. Widjaja, "ANFIS-based Fingerprint Matching Algorithm", Optical Engineering, SPIE-International Society for Optical Engine, Volume: 43, Issue: 8, Aug. 2004, Pages: 1814 – 1819

[10] N. Kawaguchi, S. Matsubara, K. Takeda, and F. Itakura, "Multimedia Data Collection of In-Car Speech Communication," Proceedings 7th European Conference on Speech Communication and Technology, Sep. 2001, Pages: 2027 – 2030

[11] J. Bengtsson, R. Johansson, and A. Sjogren, "Modeling of Drivers' Behavior", Proceedings of Advanced Intelligent Mechatronics. 2001 IEEE/ASME International Conference, Volume: 2, 8-12 July 2001 Pages: 1076 - 1081

[12] H. Ohno, "Analysis and Modeling of Human Driving Behaviors Using Adaptive Cruise Control", Industrial Electronics Society 2000, IECON 2000. 26th Annual Conference of the IEEE, Volume: 4, 22-28 Oct. 2000, Pages: 2803 - 2808

[13] M. Chan, A. Herrera, and B. Andre, "Detection of changes in driving behaviour using unsupervised learning", Systems, Man, and Cybernetics, 1994 IEEE International Conference on Humans, Information and Technology, Volume: 2, 2-5 Oct. 1994, Pages: 1979 - 1982

Chapter 3

DRIVER IDENTIFICATION BASED ON SPECTRAL ANALYSIS OF DRIVING BEHAVIORAL SIGNALS

Yoshihiro Nishiwaki[1], Koji Ozawa[1], Toshihiro Wakita[2], Chiyomi Miyajima[1], Katsunobu Itou[1], and Kazuya Takeda[1]

[1]Graduate School of Information Science, Nagoya University, Nagoya 464-8603, JAPAN; [2]Toyota Central R&D Labs., Yokomachi, Nagakute, Aichi, 480-1192, JAPAN

Abstract: In this chapter, driver characteristics under driving conditions are extracted through spectral analysis of driving signals. We assume that characteristics of drivers while accelerating or decelerating can be represented by "cepstral features" obtained through spectral analysis of gas and brake pedal pressure readings. Cepstral features of individual drivers can be modeled with a Gaussian mixture model (GMM). Driver models are evaluated in driver identification experiments using driving signals of 276 drivers collected in a real vehicle on city roads. Experimental results show that the driver model based on cepstral features achieves a 76.8 % driver identification rate, resulting in a 55 % error reduction over a conventional driver model that uses raw gas and brake pedal operation signals.

Key words: Driving behavior, driver identification, pedal pressure, spectral analysis, Gaussian mixture model

1. INTRODUCTION

The number of driver license holders and car owners are increasing every year, and the car has obviously become indispensable in our daily lives. To improve safety and road traffic efficiency, intelligent transportation system (ITS) technologies including car navigation systems, electronic toll collection (ETC) systems, adaptive cruise control (ACC), and lane-keeping assist systems (LKAS) have been developed over the last several years. ACC

and LKAS assist drivers by automatically controlling vehicles using observable driving signals of vehicle status or position, e.g., velocity, following distance, and relative lane position. Other efforts addressing driving signals include driving behavior modeling that predicts the future status of a vehicle [1] [2], drowsy or drunk driving detection based on eye-monitoring [3] [4], and the cognitive modeling of drivers [5]. Driving behaviors are different among drivers, and as such, modeling of drivers' characteristics in driving behaviors has also been investigated for intelligent assistance for each driver [6] [7]. In [6] and [7], drivers were modeled using Gaussian mixture models (GMMs) [8] that characterized the distributions of gas and brake pedal pressure, velocity, and following distance.

In this research, we have focused on characteristics of drivers from the driving behavior of their gas and brake pedal operation. We have applied cepstral analysis to the gas and brake pedal signals to obtain cepstral coefficients, which are the most widely used spectral features in the speech recognition community. From a theoretical point of view, a cepstrum is defined as the inverse Fourier transform of the log power spectrum of the signal, which allows us to smooth the structure of the spectrum by keeping only the first several low-order coefficients and setting the remaining coefficients to zero. Cepstral coefficients are therefore convenient for representing the spectral envelope.

Assuming that driver characteristics under driving conditions while accelerating or decelerating could be represented by spectral envelopes of pedal operation signals, we have modeled the characteristics of each driver with a GMM using the lower-order cepstral coefficients. GMM driver models based on cepstral features were evaluated in the identification of 276 drivers and compared to conventional GMM driver models that used raw driving signals without any applied spectral analysis techniques.

2. DRIVING BEHAVIORAL SIGNALS

2.1 Driving Signals

Observable driving signals can be categorized into three groups:

1. Driving behavioral signals, e.g., gas pedal pressure, brake pedal pressure, and steering angle.
2. Vehicle status signals, e.g., velocity, acceleration, and engine speed.
3. Vehicle position signals, e.g., following distance, relative lane position, and yaw angle.

Among these signals, we focused here on the driving behavioral signals, especially on the drivers' characteristics with respect to gas and brake pedal pressures.

2.2 Data Collection

Driving behavioral signals were collected using a data collection vehicle (Toyota Regius), which has been specially designed for data collection in the Center for Integrated Acoustic Information Research (CIAIR) project at Nagoya University, Japan. Detailed information on this corpus can be found in [9] and in Chapter 1 of the first volume in this series [11]. Each driver drove the car on a city road, and five-channel driving signals as well as 16-channel speech signals, three-channel video signals, and GPS were recorded. The driving signals included pressure on gas and brake pedals, engine speed, car velocity, and the steering angle. These signals were originally sampled at 1 kHz and down-sampled to 100 Hz in experiments.

Figure 3-1 shows examples of three-minute driving behavioral signals collected in the vehicle. Figures correspond to the force on gas pedal (top) and brake pedal (bottom), respectively.

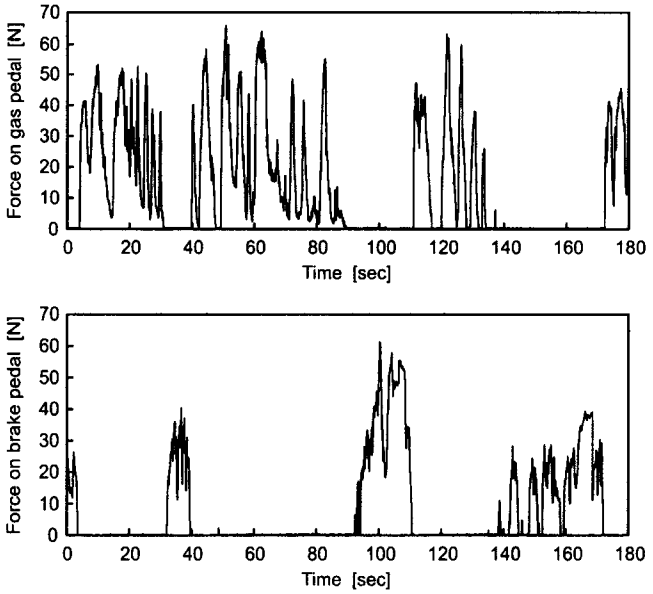


Figure 3-1. Examples of driving behavioral signals (Top: gas pedal pressure; Bottom: brake pedal pressure).

3. DRIVER MODELING

3.1 Spectral Analysis of Pedal Signals

Examples of gas pedal operation signals for two drivers are shown in Fig. 3-2 (left) and their corresponding spectra are shown in Figure 3-2 (right). Each figure shows three examples of 0.32-second long gas pedal signals. Driver A in Figure 3-2 (top) tends to increase the pressure on the gas pedal gradually, whereas driver B in Figure 3-2 (bottom) accelerates in two stages. After the initial acceleration, driver B momentarily reduces the pressure on the gas pedal, and then resumes acceleration.

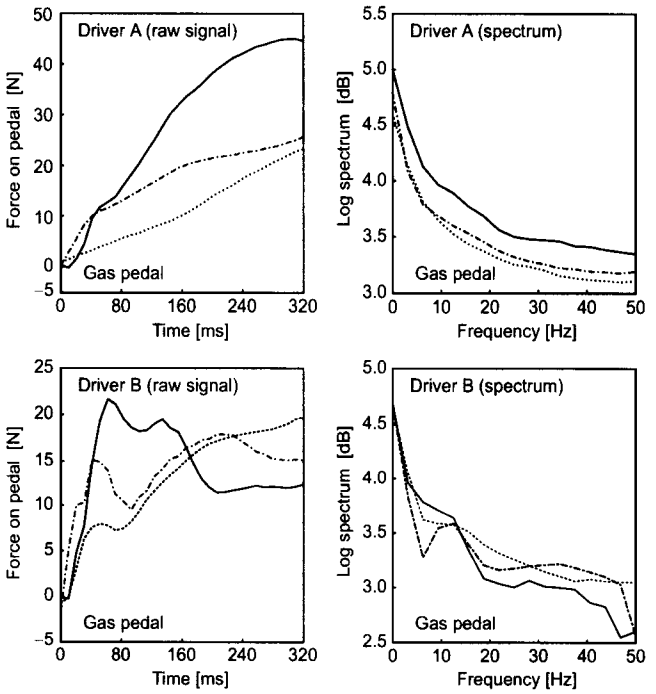


Figure 3-2. Gas pedal signals (left) and their spectra (right) (Top: driver A; Bottom: driver B).

We can see that the spectra shown in the right figures are similar for the same driver but different between the two drivers. Assuming that the spectral envelope can capture the differences between the characteristics of among different drivers, we focused on the differences in spectral envelopes represented by cepstral coefficients.

3.2 GMM Driver Modeling and Identification

A Gaussian mixture model (GMM) [8] was used to represent the distributions of feature vectors of cepstral coefficients of each driver. The GMM parameters were estimated using the expectation maximization (EM) algorithm. The GMM driver models were evaluated in driver identification experiments, in which the unknown driver was identified as driver k^* who gave the maximum weighted GMM log likelihood over gas pedal and brake pedals:

$$k^* = \arg\max_k \left\{ \alpha \log P\left(\mathbf{G}\middle|\lambda_{G,k}\right) + (1-\alpha) \log P\left(\mathbf{B}\middle|\lambda_{B,k}\right) \right\}, \ 0 \le \alpha \le 1 \quad (1)$$

where $\mathbf{G}$ and $\mathbf{B}$ are the cepstral sequences of gas and brake pedals, and $\lambda_{G,k}$ and $\lambda_{B,k}$ are the k-th driver models of gas and brake pedals, respectively; α is the linear combination weight for the likelihood of gas pedal signals.

4. DRIVER IDENTIFICATION EXPERIMENT

4.1 Experimental Conditions

Table 3-1. Experimental conditions.

Number of drivers	276
Training data length	3 min
Test data length	3 min
Sampling frequency	100 Hz
Frame length	0.32 sec
Frame shift	0.1sec
Analysis window	Rectangular window
Number of Gaussians	8, 16, 32
Cepstral coefficients	c(0) - c(15)
Δ window length	0.8 sec
Weight for gas pedal likelihood α	0 - 1

Table 3-1 displays experimental conditions for driver identification. We used driving data from 276 drivers who drove for more than six minutes, excluding the data gathered while not moving. The driving signals of the first three minutes were used for training, and the second three minutes for the test. We modeled the distribution of cepstral coefficients and their dynamic features (Δ coefficients) using GMMs with 8, 16 or 32 Gaussians and diagonal covariance matrices.

As in the case of speech recognition, we also use the dynamic features of the driving behavioral signals defined as linear regression coefficients:

$$\Delta x(t) = \frac{\sum_{k=-K}^{K} kx(t+k)}{\sum_{k=-K}^{K} k^2} \tag{2}$$

where $x(t)$ is the raw signal at time t, and K is the half window size for calculating the Δ coefficients. We have selected $2K = 800$ ms as the best window size from preliminary experiments. Frame length, frame shift, and the range of cepstral coefficients were also determined in the preliminary experiments.

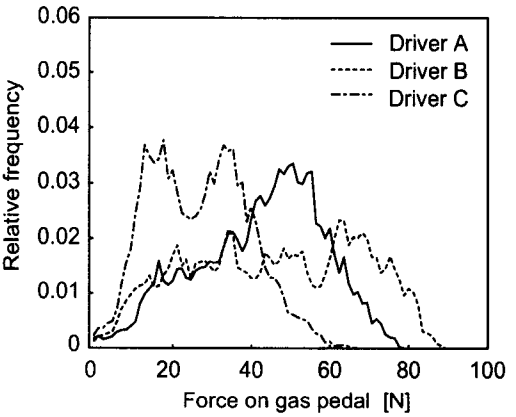


Figure 3-3. Distribution of raw signal.

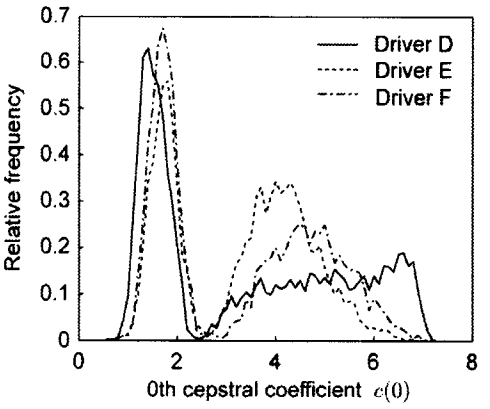


Figure 3-4. Distribution of the zeroth cepstral coefficient.

We have also compared the driver models based on cepstral features to the conventional driver models based on the raw driving signals. Examples of distributions of raw gas pedal signals are shown in Figure 3-3 and distributions for the zeroth cepstral coefficient are given in Figure 3-4. Significant differences in distributions among drivers can be observed in both figures.

4.2 Experimental Results

Figures 3-5 and 3-6 show identification results for GMM driver models with 8, 16, and 32-components using raw signals and cepstral coefficients (cepstrum), respectively. The leftmost results correspond to the identification rates when using only the brake pedal signals, and rightmost results were obtained with gas pedal signals alone. We can see that the gas pedal signals gave better performance than brake pedal signals. This is because drivers hit the gas pedal more frequently than the brake pedal as shown in Fig. 1.

The results for 16-component GMM in Figures 3-5 and 3-6 are summarized in Figure 3-7. The identification performance was rather low when using the raw driving signals: the best identification rate for raw signals was 47.5 % with $\alpha = 0.80$. By applying cepstral analysis, however, the identification rate increased to 76.8 % with $\alpha = 0.76$. We can thus conclude that cepstral features could capture the individualities in driving behavior better than raw driving signals, and could achieve better performance in driver identification. We also carried out driver identification experiments using driving signals collected on a driving simulator, obtaining similar results [10].

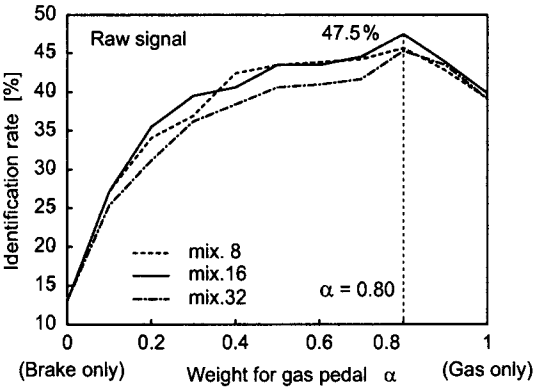


Figure 3-5. Results for combination of gas and brake pedal (raw signals).

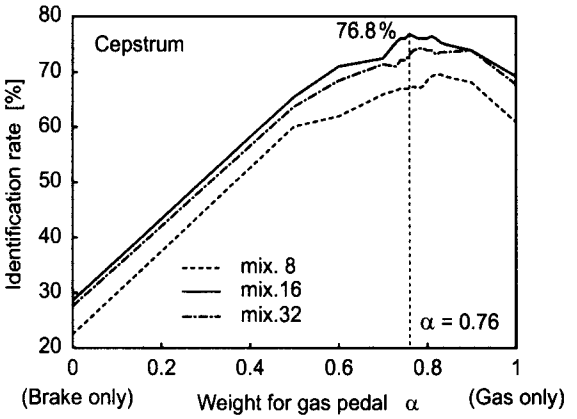


Figure 3-6. Results for combination of gas and brake pedal (cepstral coefficients).

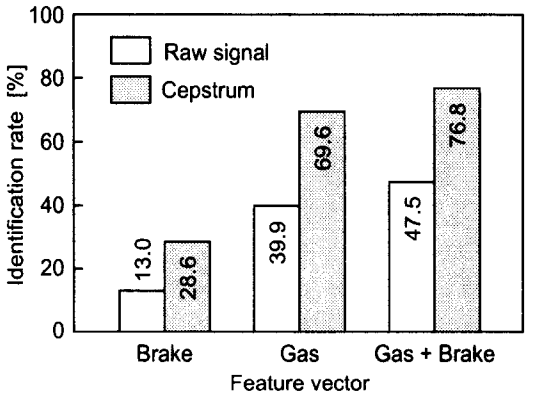


Figure 3-7. Comparison of the identification rate between the conventional model and the proposed model.

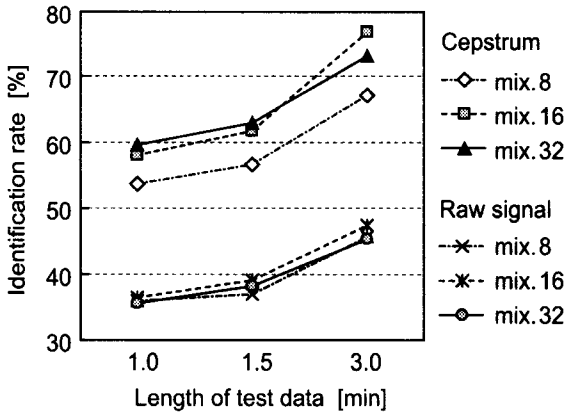


Figure 3-8. Results for different test lengths.

4.3 Experiments on Different Test Lengths

We have investigated driver identification performance for different testing conditions. Figure 3-8 shows identification rates when changing the test data length as 1, 1.5, and 3 minutes long. Although the identification rate for cepstral features has deteriorated to 59.5 % with a one-minute test data, it still performed better than the identification rate of raw signals with a three-minute test data.

5. CONCLUSION

In this chapter, we have investigated the modeling of individuality from driving behavioral signals. We have modeled the distribution of cepstral coefficients of gas and brake pedal operation signals using the property that the spectral envelopes are similar for the same driver but different among different drivers. Driver models were evaluated in driver identification experiments. Using cepstral features we have achieved an identification rate of 76.8 % for 276 drivers, which corresponds to a 55 % error reduction over the conventional driver model based on raw pedal operation signals.

The selective use of driving signals while accelerating or decelerating and the modeling of characteristics in longer-term driving signals (more than a 0.32-second frame length) is to be addressed in our future work. Other driver modeling techniques apart from GMM, such as hidden Markov models, can be employed for more efficient modeling of the time series of feature vectors. We also plan to extend driver modeling to driver-type modeling to intelligently assisting a particular driver by clustering the drivers into certain groups (e.g., impatient, aggressive, alert, etc.).

REFERENCES

[1] A. Pentland and A. Liu, "Modeling and prediction of human behavior," Neural Computation, vol. 11, pp. 229-242, 1999.

[2] N. Oliver and A.P. Pentland, "Driver behavior recognition and prediction in a SmartCar," Proc. SPIE Aerosense 2000, Enhanced and Synthetic Vision, Apr. 2000.

[3] R. Grace, V. E. Byrne, D. M. Bierman, J. Legrand, D. Gricourt, B. K. Davis, J. J. Stastzewski, and B. Carnahan, "A drowsy driver detection system for heavy vehicles," Proc. 17th Digital Avionics Systems Conference, vol. 2, no. I36, pp. 1-8, Oct. 1998.

[4] P. Smith, M. Shah, and N. da V. Lobo, "Monitoring head/eye motion for driver alertness with one camera," Proc. ICPR 2000, vol. 4, pp. 636-642, Sept. 2000.

[5] D. D. Salvucci, E. P. Boer, and A. Liu, "Toward an integrated model of driver behavior in a cognitive architecture," Transportation Research Record, 2001.

[6] K. Igarashi, C. Miyajima, K. Itou, K. Takeda, F. Itakura, and H. Abut "Biometric identification using driving behavioral signals," Proc. IEEE-ICME 2004, TP1-2, June 2004.

[7] T. Wakita, K. Ozawa, C. Miyajima, and K. Takeda, "Parametric Versus Non-Parametric Models of Driving Behavior Signals for Driver Identification," Proc. AVBPA 2005, July 2005.

[8] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," IEEE Trans. Speech and Audio Processing, vol. 3, no. 1, pp. 72-83, Jan. 1995.

[9] N. Kawaguchi, S. Matsubara, K. Takeda, and F. Itakura, "Multimedia data collection of in-car speech communication," Proc. EUROSPEECH 2001, pp. 2027-2030, Sept. 2001.

[10] K. Ozawa, T. Wakita, C. Miyajima, K. Itou, and K. Takeda, "Modeling of individualities in driving through spectral analysis of behavioral signals," Proc. IEEE-ISSPA 2005, pp. 851-854, Aug. 2005.

[11] N. Kawaguchi, S. Matsubara, I. Kishida, Y. Irie, H. Murao, Y. Yamaguchi, K. Takeda and F. Itakura, "Construction and Analysis of the Multi-layered In-car Spoken Dialogue Corpus," Chapter 1 in *DSP in Vehicular and Mobile Systems,* H. Abut, J.H.L. Hansen, K. Takeda (Editors), Springer, New York, NY, 2005.

Chapter 4

AN ARTIFICIAL-VISION BASED ENVIRONMENT PERCEPTION SYSTEM

Application to Autonomous Vehicle Navigation in Urban Areas

S. Nogueira[1], Y. Ruichek[1], F. Gechter[1], A. Koukam[1], and F. Charpillet[2]
[1]*Systems and Transportation Laboratory F-90010 Belfort cedex*
[2]*INRIA LORIA UMR 7503 BP239 F-54506 Vandoeuvre cedex*

Abstract: Recently, many research programs have investigated the concept of intelligent vehicles and their integration in the city of tomorrow. The aim is to develop an intelligent transportation system based on a fleet of fully automated cars designed for short trips at low speed in urban areas. This system will offer advantages of high flexibility, efficiency, safety, and thus, will improve the quality of life in our cities including but not restricted to protection of the environment, better management of parking areas, and others. One of the key functions that such transportation system must achieve concerns the notion of autonomous navigation of vehicles. To reach this goal, we are working on vehicle environment perception using passive and active sensor technologies. In this chapter, we address an artificial-vision based environment perception system for autonomous vehicle navigation. We are particularly interested in obstacle detection using stereo vision, image road line tracking for vehicle road, line following and landmarks recognition for local positioning. The developed techniques are implemented and have been tested using a fully automated vehicle platform for autonomous navigation in urban areas.

Key words: Intelligent vehicle, autonomous navigation, stereovision, obstacle detection, image road line tracking, vehicle road line following, landmarks recognition

1. INTRODUCTION

One of the challenges of the research studies in transportation systems is to develop the concept of intelligent vehicles and their integration in the city of tomorrow [1–6]. The aim is to propose an intelligent system for shared and/or collective public transportation in urban areas. Based on a fleet of

fully automated cars, this system will improve the quality of life in our cities in terms of safety, noise and pollution reduction, good management of parking areas, etc. Towards that end, the Institut National de Recherche en Informatique et Automatique (INRIA), the Institut National de Recherche sur les Transports et leur Sécurité (INRETS) and the RoboSoft Industrial Society have jointly developed a fully-automated electric vehicle [7].

One of the key tasks that such vehicle must achieve is environment vehicle perception for autonomous navigation. In this study, we propose an artificial vision based environment perception system ensuring obstacle detection using stereo vision, image road line tracking for vehicle road line following and landmarks recognition for local positioning.

The chapter is organized as follows. Section 2 presents our experimental vehicle. Section 3 describes the stereo vision based obstacle detection technique. The image road line tracking and vehicle road line following methods are detailed in section 4. Section 5 presents the proposed landmarks recognition technique. Section 6 concludes the paper with some current research works.

2. EXPERIMENTAL VEHICLE PRESENTATION

Our experimental vehicle in Figure 4-1 has four driving and steering wheels. Each wheel has its own electrical motor and each pair of wheels is controlled by a MPC555 (Motorola PowerPC Processor) card. The two MPC555 cards are managed by an embedded PC with an X86 processor through a CAN bus. The user interface is ensured by a tactile screen. The user can also access to the control system by a WI-FI communication ability.

As the control architecture is distributed, the low level programming tools are based on a particular software called SynDEX (System-level CAD Software for Distributed Real-Time Embedded Systems) [8]. To perceive the environment of the vehicle, we use a second embedded PC, which is dedicated to data acquisition and processing. The information exchange between the two PCs is ensured by socket programming.

The vehicle is equipped by several sensors: two stereo vision sensors, laser range finder, sonar sensors, magnetic field sensors, and a Global Positioning System (GPS).

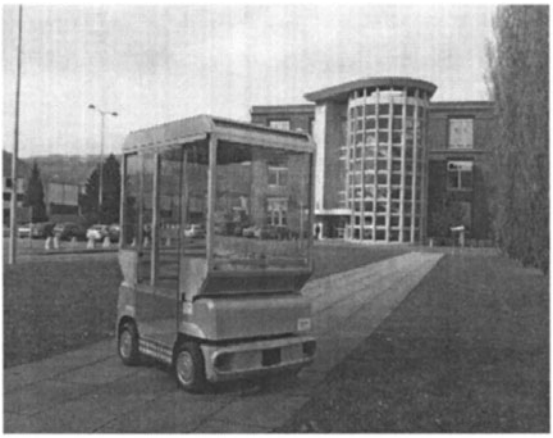


Figure 4-1. Experimental fully automated vehicle.

3. STEREO VISION BASED OBSTACLE DETECTION

Stereo vision is a known approach for recovering 3-D information of a scene seen by two or more video cameras from different viewpoints. The difference of the viewpoint positions in the stereo vision system causes a relative displacement, called disparity, of the corresponding features in the stereo images. This relative displacement encodes the depth information, which is lost when the three dimensional structure is projected on a retinal plane. The key problem is the stereo matching task, which consists of comparing each feature extracted from one image with a number of – generally large– features extracted from the other image in order to find the corresponding one, if any. In addition to being difficult to perform and computationally extensive, this process requires a large amount of memory. Once the matching is established and the stereo vision system parameters are known, the depth computation is reduced to a simple triangulation technique.

3.1 Feature Extraction

The low-level processing of a couple of two stereo images yields the features required in the correspondence phase. Edges are valuable candidates for matching because large local variations in the gray-level function correspond to the boundaries of objects being observed in a scene.

Considering the lines and columns of an image, the edge detection is performed by means of the Deriche's operator [9]. After derivation, the

pertinent local extrema are selected by splitting the gradient magnitude signal into adjacent intervals where the sign of the response remains constant (cf. Figure 4-2). In each interval of constant sign, the maximum amplitude indicates the position of an unique edge associated to this interval when, and only when, this amplitude is greater than a low threshold value t [10]. The application of this threshold procedure allows removing non significant responses of the differential operator lying in the $[-t,+t]$ range. The adjustment of t is not crucial. Rather encouraging results have been obtained with t adjusted at 10% of the greatest amplitude of the response of the differential operator.

Applied to the left and right stereo images, this edge extraction procedure yields two lists of edges. Each edge is characterized by its position in the image, the amplitude, the sign and the orientation of the gradient.

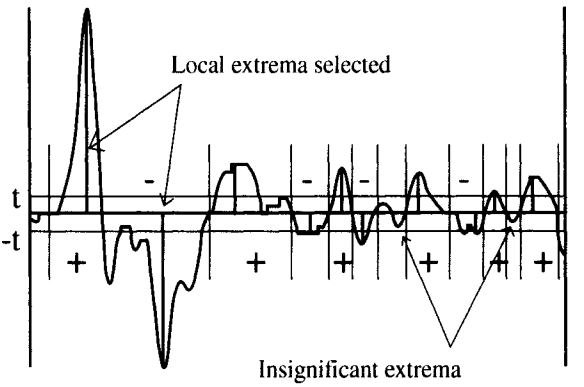


Figure 4-2. Edge extraction from the derivative of a line or column of an image.

3.2 Edge Stereo Matching

For real-time obstacle detection, we have developed a fast stereo matching method based on a voting strategy [11]. The matching procedure is applied to a set of epipolar lines and uses only possible pairs which respect three local constraints: position, slope and orientation constraints. The searching of the best matches is based on three global constraints: uniqueness, ordering and smoothness constraints. The searching procedure consists in assigning for each possible match a score, which represents a quality measure of the matching with respect to the global constraints.

Figure 4-3 represents the matching matrix in which we consider only the possible matches (white circles). Let M_{lr} be an element of the matching matrix, representing a possible match between the edges l and r in the left and right images, respectively.

The stereo matching procedure starts by determining among the other possible matches the ones that are authorized to contribute to the score SM_{lr} of the possible match M_{lr} The contributor elements are obtained by using the uniqueness and ordering constraints: an element $M_{l'r'}$ is considered as a contributor to the score of the element M_{lr} if the possible matches (l,r) and (l',r') verify the local constraints. The contributors of the score of the element M_{lr} are the possible matches lying in the gray area of the Figure 4-3. The contribution of the contributors is then performed by means of the smoothness constraint. For each contributor $M_{l'r'}$, the score updating rule is defined as follows:

$$SM_{lr}(new) = SM_{lr}(old) + f(X_{lrl'r'}) \tag{1}$$

where $X_{lrl'r'}$ is the absolute value of the difference between the disparities of the pairs (l,r) and (l',r'), with $f(X) = (1+X)^{-1}$. Finally, a procedure is designed to determine the correct matches by selecting in each row of the matching matrix the higher score element. More than one element can be selected in the same column. To discard this situation, which corresponds to multiple matches, we apply the same procedure to each column of the matching matrix. The elements selected by this two-step procedure indicate the correct matches.

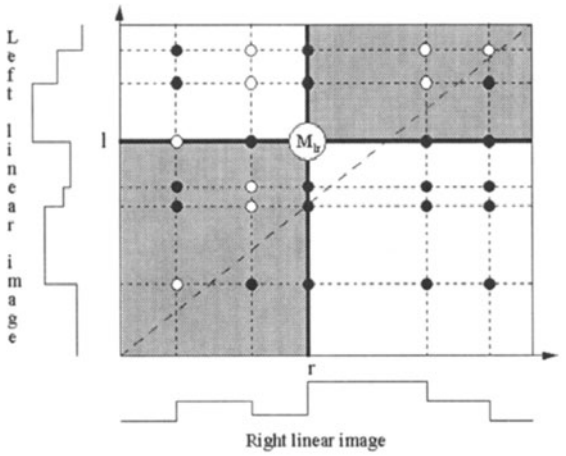


Figure 4-3. Matching matrix.

4. LINE TRACKING AND FOLLOWING

The vehicle autonomous navigation is based on road line following, which is performed by means of image road line detection and tracking.

4.1 Base Road Line Detection

To determine the base road line, the processing involves the bottom of the first frame, starting by the last line. The processing of a line consists of searching two consecutive opposite local variations of the gray level intensity ΔI_i and ΔI_j, for which the distance between their positions i and j is close to the width of the projection of the road line on an image. To reduce the solution space, we consider only local variations greater than a low threshold value *TH*. This thresholding procedure allows discarding insignificant local variations. Our searching procedure can be expressed as follows:

$$|\Delta I_i| > TH \text{ and } |\Delta I_j| > TH \tag{2}$$

$$\Delta I_i \cdot \Delta I_j < 0 \tag{3}$$

$$|j - i - W| < \varepsilon \tag{4}$$

where i and j are the position of two successive local variations in a line of an image, with $i < j$ and W represents the width of the road line on an image projection and ε allows a tolerance.

If there are more than one solution we retain only the solution for which the positions of the two successive local variations are close to the middle of the line of the image. Indeed, the goal of the road line following is to keep the projection of the road line in the middle of the image.

4.2 Road Line Detection

Once the base of the line is detected, the next step is to detect the whole line in the whole image. At this stage, we suppose that base road line detection is applied successfully to at least two lines of the bottom of the first frame. The line detection starts from the left points composing the base image road line. Let P_n and P_{n+1} be the last successive points belonging to the base image road line. Let P_{n+2} the next point to determine. For a fast and reliable detection, the considered searching space we consider is composed

by the seven points $\{Q_i\}_{i=0 \text{ to } 6}$ belonging to the top half discrete circle defined by P_{n+1} as a center (cf. Figure 4-4).

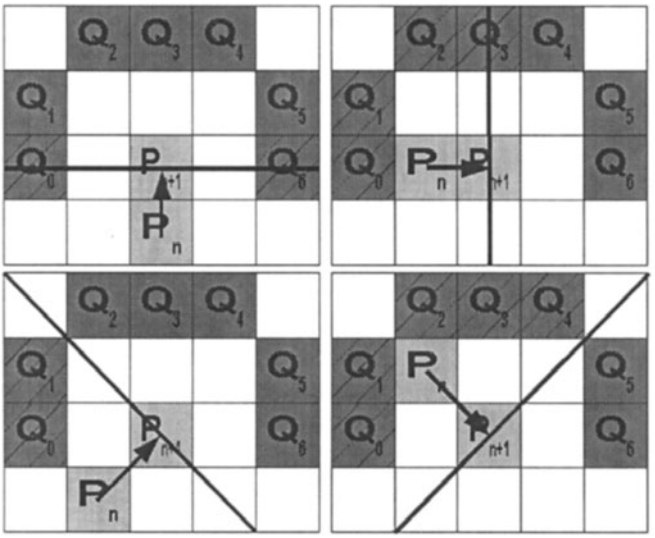


Figure 4-4. Road line detection.

The searching of the point P_{n+2} is based on the maximization of the local variation of the gray level intensity. First, from the points $\{Q_i\}_{i=0 \text{ to } 6}$, we keep only those respecting the direction constraint:

$$\vec{u} \cdot \vec{v} > 0 \tag{5}$$

where $\vec{u} = (P_{n+1} - P_n)$ and $\vec{v} = (Q_i - P_{n+1})$. We select then, from the remaining points, the one with the maximum local variation of the gray level intensity. This selected point is hence P_{n+2}.

4.3 Road Line Tracking

Road line tracking consists of determining the points of the road line in the current frame (t+1) from the points of the road line in the previous frame (t). Let $P_{xy}(t)$ be a point of the road line in the previous frame. Let $\Omega^k_{xy}(t)$ an horizontal local neighbor of the point $P_{xy}(t)$, defined as follows:

$$\Omega^k_{xy}(t) = \{P_{x'y} / x - k \le x' \le x + k\} \tag{6}$$

where k is a positive constant, chosen according to the vehicle dynamic. This neighbor is used to detect a point of the road line in the current frame. This point corresponds to the maximum local variation of the gray level intensity in $\Omega_{xy}^{k}(t)$. Figure 4-5 shows the result of the road line tracking procedure. It is worth noting that the proposed road line detection and tracking is adapted successfully to analyze dashed road lines.

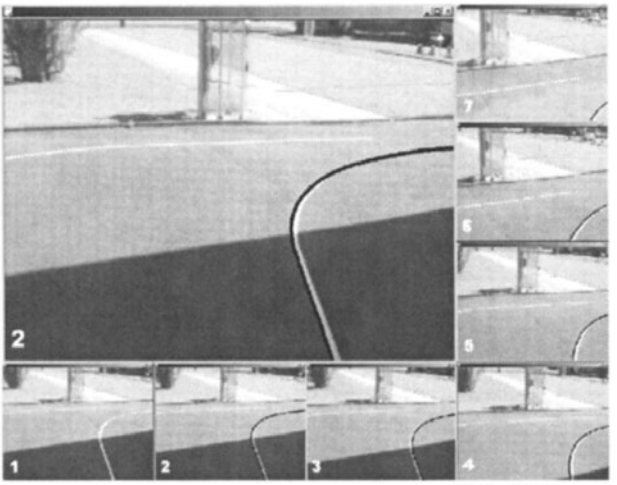


Figure 4-5. Image road line tracking.

Subject to conditions of the environment, the tracking procedure may miss some points in the current frame. If the number of detected points is under a given threshold, we apply again the detection road line procedure in the whole image representing the current frame. When the number of detected points is insufficient to get a good detection, the car stops unless another guidance system takes place.

4.4 Vehicle Road Line Following

Vehicle road line following is based on the image road line tracking. It consists of keeping the projection of the road line in the middle of the acquired images. Because of the non-visible area in front of the vehicle due to camera position (cf. Figure 4-6), a simple following procedure fails by creating a shift between the vehicle and the road line. This situation occurs when the road line presents curves. As a consequence, the procedure will be ineffective because the road line will disappear from the images.

To overcome this problem, the approach we propose is based on the recording of the vehicle's trajectory at regular distance interval. The vehicle road line following is performed thanks to a fuzzy logic system, which

estimates the direction of the vehicle's wheels from the vehicle's speed and the current shift between the vehicle and the road line. Figure 4-7 shows the fuzzy input and output for which the fuzzy rules are illustrated in Table 4-1. To solve the fuzzy system, we use the Min/Max inference and the center of gravity for the defuzzification step.

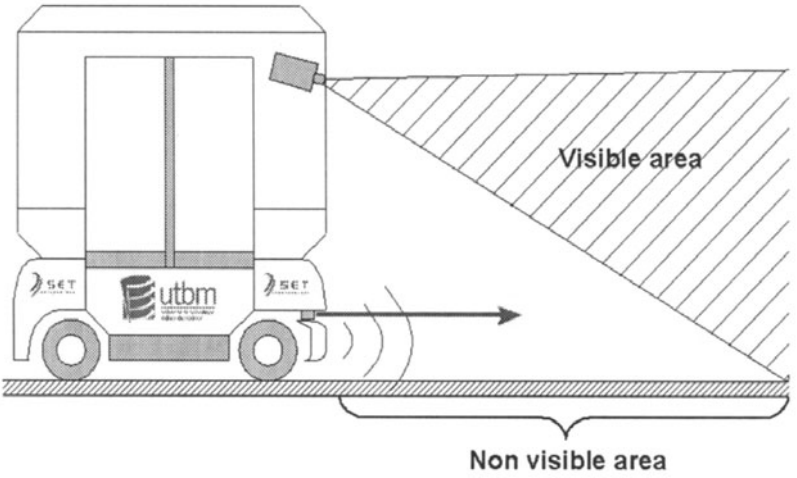


Figure 4-6. Non-visible area.

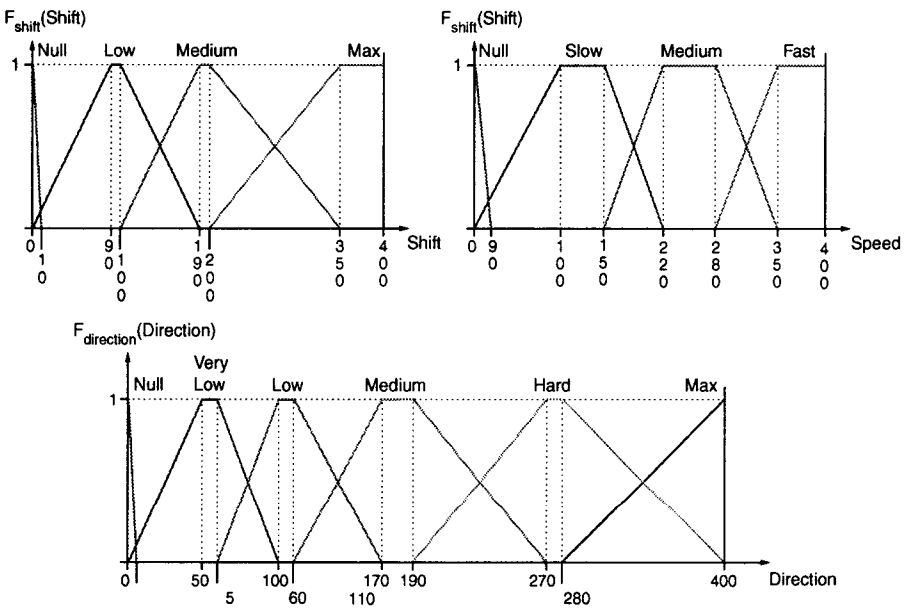


Figure 4-7. Fuzzy input and output.

Table 4-1. Fuzzy Associative Matrix (FAM).

AND	*Null*	*Low*	*Medium*	*Max*
Null	Null	Low	Hard	Max
Slow	Null	Low	Hard	Hard
Medium	Null	Very Low	Medium	Medium
Fast	Null	Very Low	Low	Low

5. LANDMARKS RECOGNITION

In order to get some information from the vehicle surrounding as relative position of the vehicle, we have developed a landmarks recognition technique. We use P-similar landmarks [12], which have particular shape allowing real-time recognition (cf. Figure 4-8). Furthermore, they are easy and economical to install in vehicles.

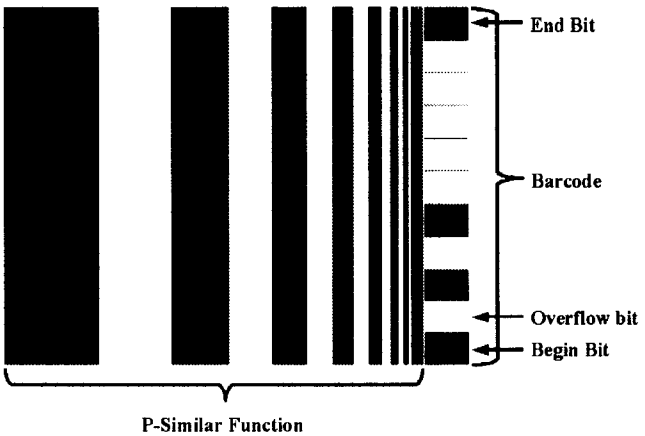


Figure 4-8. P-similar landmark.

5.1 P-Similar Landmarks Generation and Detection

A P-similar function f is defined as [13]:

$$\begin{cases} f:\Re^{+}\to\Re \\ \exists p\in\left]0;1\right[,\forall x>0, f(x)=f(p\cdot x) \end{cases} \tag{7}$$

In our application, we use square P-similar landmarks, which are generated by square P-similar functions *S*, defined as follows:

$$\forall x > 0, S_p(x) = \left\lfloor \left(2 * \left(\log_p(x)\right) - \left\lfloor \log_p(x) \right\rfloor\right)\right\rfloor \tag{8}$$

To detect a square P-similar landmark, we compute for a pixel, defined by its position (x,y) in the image, the following quantity:

$$m_y(x) = \frac{1}{w} * \sum_{j=0}^{w} \left[\begin{array}{l} \left| I(x+j,y) - I\left(x + j\sqrt{p}, y\right)\right| - \\ \left| I(x+j,y) - I(x + p \cdot j, y)\right| \end{array} \right] \tag{9}$$

where I is the gray-level intensity function and w is the width of the window processing. The quantity $m_y(x)$ takes three possible values: 1, -1 or 0. When $m_y(x) = 1$, a square P-similar landmark is detected, when $m_y(x) = -1$, a square $\sqrt{p}$ -similar landmark is detected, when $m_y(x) = 0$, no P-similar landmark is detected. Note that P-similar and $\sqrt{p}$ -similar are equivalent: P-similar can be used to detect far P-similar landmarks, whereas $\sqrt{p}$ -similar can be used to detect close P-similar landmarks.

5.2 Barcode for Landmarks Identification

Barcodes are used to identify the P-similar landmarks (cf. Figure 4-8). The barcode is composed of 11 bits. 3 bits are necessary for the encoding identification: the "Begin Bit" indicates the beginning of the code, the "End Bit" indicates the end of the code and the "Overflow Bit". The useful information of the barcode is encoded within the 8 remaining bits, where a black bit is set to 1 and a white bit is set to 0. With this barcode we can code $2^8 = 256$ different landmarks. If more codes are necessary, we have just to add more bits within the barcode.

6. CONCLUSION

In this study, we have presented an artificial-vision based environment perception system for autonomous navigation of vehicles in urban areas. The proposed system ensures obstacle detection using stereo vision, road line following by image, road line tracking and artificial landmarks recognition. The techniques under study have been integrated and tested using an experimental fully automated vehicle. Our current research work is focused on fusion of information from multi-sensor signals to improve the perception

techniques using passive and active technologies. We are working also on global localization by combining visual geographical data and scene analysis information. Our approach consists of comparing real images, representing the environment of the vehicle, with images generated in real-time by a Geographical Information System (GIS).

ACKNOWLEDMENT

The French Comte Regional Council is gratefully acknowledged for its financial support. The authors extend their appreciation to L. Moalic.

REFERENCES

[1] Parent M. and Fauconnier S., "Design of an electric vehicle specific for urban transport," in Congrès EVT'95. Paris, 1995.

[2] Dumontet F. Allal C. and Parent M., "Design tools for public cars transportation systems," in Fourth International Conference on Applications of Advanced Technologies in Transportation Engineering. Capri, Italy, 1995.

[3] Neurrière J.P. Augello D., Benéjam E. and Parent M., "Complementarity between public transport and a car sharing service," in First World Congress on Applications of Transport Telematics & Intelligent Vehicule-Highway Systems, 1994.

[4] Texier P.Y. Parent M., Dumontet F. and Leurent F., "Design and implementation of a public transportation system based on self-service electic cars," in IFAC/IFORS Congress. Tianjin, China, 1994.

[5] Mobivip, http://www-sop.inria.fr/mobivip

[6] CyberC3, http://CyberC3.sjtu.edu.cn/

[7] Robosoft, http://www.robosoft.fr

[8] System level CAD software for optimizing distributed real-time embedded systems. Journal ERCIM News, October 2004,

[9] Deriche R., "Fast algorithms for low-level vision," in IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 12, no. 1, 1990.

[10] Ruichek Y., Postaire J.G., and Bruyelle J.L., "Detecting and localising obstacles in front of a moving vehicle using linear stereo vision," in Mathematical and Computer Modelling, vol. 22, no. 4-7 235-246, 1995.

[11] Hariti M., Ruichek Y., and Koukam A., "A voting stereo matching method for real-time obstacle detection," in In Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003, pp. 1700–1704.

[12] Scharstein D., "Fast recognition of self-similar landmarks," in Perception for Mobile Agents (in conjunction with IEEE CVPR 99), 1999.

[13] Scharstein D. and Briggs A. J., "Fast recognition of self-similar landmarks," in On Workshop on Perception for Mobile Agents, 1999, pp. 74–81.

Chapter 5

VARIABLE TIME-SCALE MULTIMEDIA STREAMING OVER 802.11 INTER-VEHICLE AD-HOC NETWORKS

Antonio Servetti, Enrico Masala, Paolo Bucciol, and Juan Carlos De Martin
Dipartimento di Automatica e Informatica - Politecnico di Torino - Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Abstract: This chapter presents an analysis of multimedia streaming in an inter-vehicle network based on 802.11b wireless devices. In such a scenario, characterized by strong link availability variations, we investigate the performance of an adaptive packet scheduling policy that adapts the inter-packet transmission interval to the channel conditions. Network simulations are used to evaluate the effects of varying the transmission time scale from zero, when the connection is not available, to as fast as possible when the channel is available and reliable. Results show that the proposed approach ensures high quality multimedia streaming among the nodes of the inter-vehicle network by heavily reducing the percentage of lost packets with only a limited increase in the delay and jitter.

Key words: Ad-hoc wireless networks; audio and video streaming; inter-vehicle.

1. INTRODUCTION

IEEE 802.11 Wireless LAN (WLAN) products have become widely used because of their simple set-up and moderate cost. Potential uses of such equipment range from WLAN hot spots to direct connectivity of devices in ad-hoc mode.

Ad-hoc networks are a key factor in the evolution of wireless communications enabling data exchange between wireless hosts in absence of a centralized fixed infrastructure. In an inter-vehicle scenario, for

instance, vehicles can operate as a pure ad-hoc network in which each individual vehicle broadcasts data to other vehicles.

Due to the relative novelty of the application, some effort have been devoted to study and simulate 802.11 inter-vehicle transmissions[1,2], and, to the best of our knowledge, exploitation of inter-vehicle 802.11 communications for real-time multimedia services received even less attention. In Bucciol[3] the performance of video communications has been evaluated while driving two cars equipped with 802.11b standard devices in urban and highway scenarios. The experiments show that each scenario presents peculiar characteristics in terms of average link availability and SNR which can be exploited to develop more efficient inter-vehicle applications. The strong link availability variations experienced in the highway scenario suggest, in fact, that a *variable time-scale transmission policy* may be investigated for multimedia streaming to mitigate the effect of frequent disconnection between the mobile stations.

Streaming implementations developed and tuned for wired connections or wireless environments with limited mobility are usually designed to cope only with limited variations in network latency and bandwidth[4]. Before starting the playout, a pre-roll delay is then used to fill the receiver buffer. Buffering, in fact, reduces system sensitivity to short-term fluctuations in the data arrival rate by absorbing variation in end-to-end delay. However, if the rate offered by the channel falls below that of the source, the buffer will soon underflow. In this case rate adaptive algorithms are used to adapt the source rate to the current state of the network so as to generate only the bandwidth that the network is capable of carrying. The assumption is, in fact, that the distortion introduced by lowering the source coding rate is smaller than the expectedly larger distortion due to packet losses.

To deal with the fast changing inter-vehicle wireless scenario, where the connection to other mobile nodes is frequently lost and the streaming flow is interrupted, in addition to the foregoing techniques, appropriate streaming algorithms must be implemented so that the receiver has enough data to continue the playback until the connection is re-established. Then the buffer needs to be refilled to a level that provides sufficient protection for a potential subsequent disconnection[5].

In this chapter we analyze the performance of a multimedia streaming *adaptive packet scheduling* (APS) technique that enables transmission rate changes by varying the inter-packet transmission interval instead of the size of multimedia packets. Streaming systems usually transmit frames at fixed time intervals, that is, with the same rate at which they will be decoded and presented to the user. In the proposed scheduling algorithm for multimedia streaming the packet scheduler is instead able to change its instantaneous transmission rate from zero (i.e., the transmission pauses) when the link is

not available, to as faster than real-time as the channel bandwidth allows (to refill the receiver buffer to the right size). The original playback rate, however, is not changed, and remains constant.

Most of the research work performed on the idea of changing the packet transmission schedule has been previously focused on end-to-end congestion control[6], or bandwidth smoothing for VBR video[7]. The effects on the quality for inelastic multimedia traffic[8], especially with regards to its delay and jitter constraints for real-time playback, still remain to be investigated.

End-to-end congestion control techniques are mainly based on two TCP-friendly rate control mechanisms: the rate adaptation protocol (RAP)[9] and the TCP friendly rate control (TFRC)[10,11]. Both these algorithms control the network status at the receiver and provide the sender with feedback information in order to adapt the output rate of the source to the channel available bandwidth.

The adaptive packet scheduling algorithm presented in this chapter, although based on TFRC for the evaluation of the available bandwidth, relies instead on the variation of the packet sending rate and not of the source rate for matching the network status. The effectiveness of this choice has been previously demonstrated in Masala[12], where the scenario was however limited to considering video streaming over wired Internet connections. Here we consider a wireless scenario where actual measurements from inter-vehicle transmissions are used to drive network simulations that analyze the performance of the adaptive packet scheduling approach in presence of high channel loss rates.

This chapter is organized as follows. Section 2 presents the adaptive packet scheduling policy and its application to the vehicular scenario. Network simulation setup and results are described in Section 3. Finally conclusions are drawn in Section 4.

2. ADAPTIVE PACKET SCHEDULING

Streaming of real-time multimedia heavily depends on timing constraints. Audio data, for example, must be played out continuously, so, if the data does not arrive in time to the client, the playout process must pause, with annoying effects for the human listener.

Before transmission over packet networks the compressed multimedia signal is divided into frames, with the property that all the data belonging to a single frame is played back at the same time during the decoding process. Each frame may be encapsulated in one or more packets that are then transmitted and stored at the receiver before decoding. All the frames must be received before their presentation time in order to decode the data without

errors. Since the Internet introduces time-varying delays, a buffer is usually employed at the receiver to provide continuous playout, while at the sender frames are generally transmitted on the network with the same rate at which they will be decoded and presented to the user.

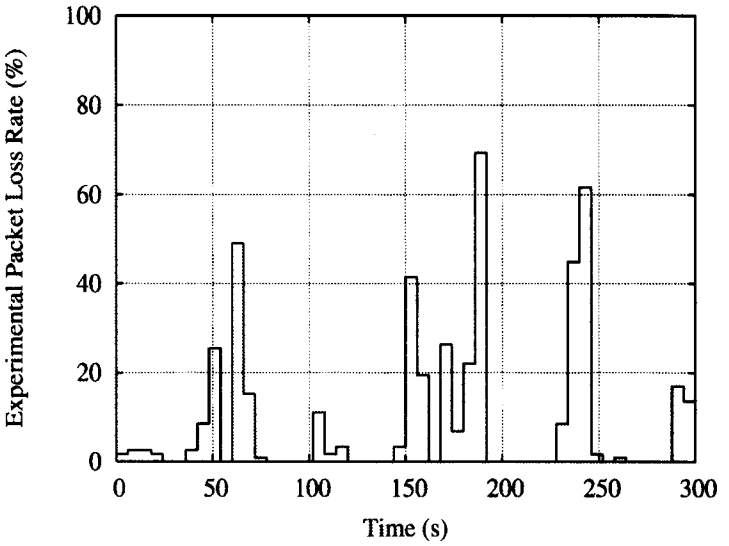


Figure 5-1. Experimental packet loss rate measured transmitting UDP packets between two vehicles on a highway. Values are averaged on a six-second window.

In the inter-vehicle scenario considered in this work, a typical problem is the possible outage of the network connection between vehicles. Such channel outages may occur while the vehicles cross areas characterized by severe fading conditions or when vehicles are at a distance close to the limit of their wireless antenna coverage area. This is a critical issue when dealing with streaming services, which may experience long (several seconds) interruption with highly negative impairments on the multimedia quality at the decoder. For instance, Figure 5-1 illustrates the packet loss rate experienced in a highway scenario, as measured in transmission experiments performed during the activity reported in Bucciol[3].

We argue that, rather than increasing the size of the playout buffer to allow additional caching and pre-fetching of multimedia data, an adaptive technique is better suited to reduce the impact of channel degradation on the perceived quality. Instead of transmitting the multimedia frames always at the same rate, our adaptive packet scheduling technique is based on the concept of varying the transmission rate according to the instantaneous network conditions, while the original playback rate remains constant.

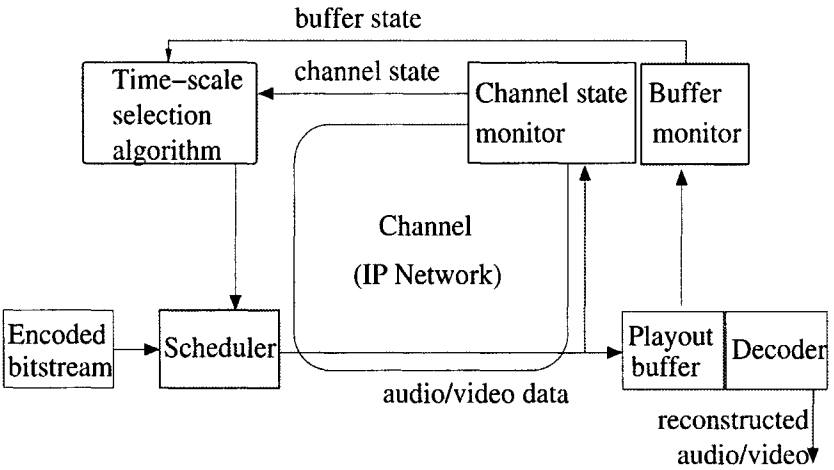


Figure 5-2. Block diagram of the adaptive packet scheduling transmission system.

Figure 5-2 depicts the block diagram of a video transmission system using the adaptive scheduling approach. The pacing of the packets is determined by the scheduler in order to achieve the rate imposed by the time-scale selection algorithm. The channel state monitor determines the instantaneous channel state. Then the status information is sent back to the time-scale selection algorithm which modifies the sending rate according to the estimate of the channel capacity. Additionally the buffer monitor may take into account the decoder buffer status and inform the scheduling algorithm with an estimate of the buffering delay experienced by the data before playout.

Because of the feedback loop between the source and the receiver the presented technique offers two advantages with respect to fixed scheduling transmission of multimedia frames. First, when a link outage occurs or when the channel is extremely bad, the number of dropped packets is considerably lower. In fact, the receiver feedback, that informs the source of the current channel loss rate, enables the sender to reduce the sending rate and so the number of packet transmitted during high error rate periods (if feedback reports are lost, the sender automatically cuts the rate in half after a given timeout expires). Second, the transmitting station may use the same feedback indication to increase the sending rate to match the available network bandwidth when the loss rate is low, thus enabling fast refill of the playout buffer. Both rate variations are obtained by varying the inter-packet gap, IPG_n, which is linked to the transmission rate, R_n, and to the packet size, s, as follows:

$$IPG_n = \frac{s}{R_n} \tag{1}$$

TFRC is used to control the maximum bit rate which can be used by the source to achieve TCP friendly behavior. The protocol is based on the following equation

$$R = \frac{s}{RTT\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3p}{8}}\right)p\left(1+32p^2\right)} \tag{2}$$

which provides the output rate, as a function of the round-trip time, RTT, the loss rate, p, the packet size, s, and the timeout interval used to reveal losses, t_{RTO}.

In addition, monitoring the instantaneous conditions of the receiver-side playout buffer allows the sender to realize when the bandwidth is not sufficient for the actual multimedia stream, i.e., a buffer underrun is likely to occur, so it can reduce the coding rate for the amount of time necessary to overcome the temporary channel degradation. This mechanism allows to combine the adaptive scheduling mechanism with the well known rate-adaptive approach.

3. NETWORK SIMULATIONS

The inter-vehicle scenario has been studied by means of simulations performed with the NS-2 network simulator (version 2.27)[13] for an 802.11b wireless LAN at 11 Mbps. Simulations aim at evaluating the adaptive packet scheduling algorithm in a context with heavy network load and with long periods of unstable connection as measured during actual experiments driving on a highway[3]. As an example of a multimedia communication, we have used a stereo MPEG-1 Layer III CBR stream at 96 kb/s per channel that achieves CD quality playback in an error free scenario.

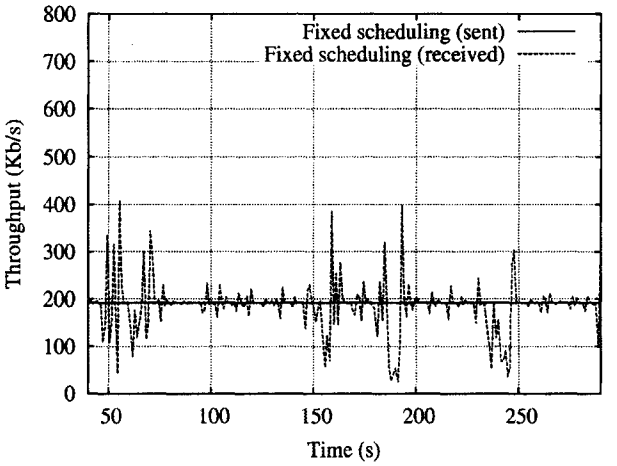


Figure 5-3. Throughput measured at the sender and at the receiver for the fixed scheduling algorithm.

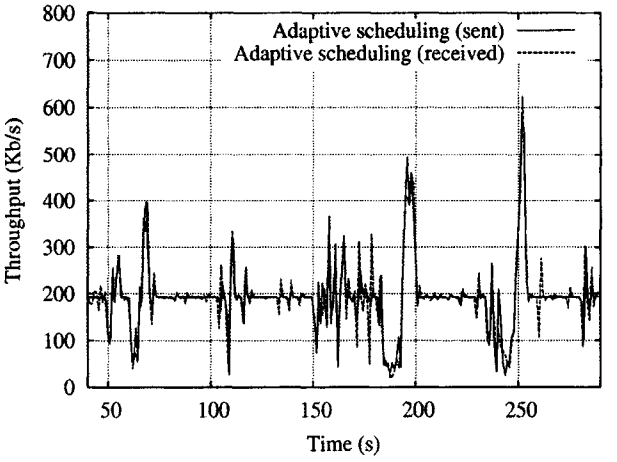


Figure 5-4. Throughput measured at the sender and at the receiver for the adaptive scheduling algorithm.

Results reported here summarize the observations made with various testing scenarios. All experiments have been carried out using a simple topology which consists of a single receiver node and a set of senders. Two nodes were instructed to send the same audio stream but with two different policies: the fixed packet scheduling (FPS) approach, in which packets are regularly spaced in time, and the adaptive packet scheduling (APS) approach in which packets are more apart from each other when the channel is bad and more closely set in time when the channel is good. The APS implementation

is based on TFRC. TFRC, in fact, can vary its sending rate in response to network congestion and its sending data rate is calculated using the TCP throughput equation. Additional interfering traffic is modeled by four nodes sending constant bit rate UDP traffic at 2.3 Mb/s in packets of 1500 bytes.

The channel behavior experienced during the actual vehicular measurements has been reproduced by means of a uniform packet error model with time varying error rate corresponding to the values shown in Figure 5-1. A number of time periods with heavy loss rate lasting several seconds are present in the experimental trace. The loss trace may exceed 50% of lost packets when the link availability between the transmitting and receiving vehicles is extremely bad. Important concerns in the evaluation of the adaptive policy are both its responsiveness to changes in network conditions and its ability to drop the packet sending rate in periods of high error rates to reduce the number of discarded packets. In particular, in the case of inelastic audio streaming, also the effect of delaying the transmission of time-sensitive data must be assessed because that practice can easily cause buffer underflows at the receiver. In the following we will present simulation results on the throughput, packet loss rate, and playout buffer fullness to validate the proposed scheduling policy with respect to the aforementioned constraints.

In Figures 5-3 and 5-4 we plot the throughput of the audio transmission as measured at both the source and the sink nodes. The key insight here is that in the APS case the number of packets sent through the network is clearly influenced by the various network conditions, i.e., the channel loss rate. The APS algorithm reduces the transmission rate by varying the inter-packet gap (IPG) around the 60th, 180th, and 230th second thus reducing the number of packet sent when they have a very high probability to be corrupted. On the other hand we can clearly note that the throughput increases well above the average bitrate of 192 kb/s just after those time periods when the sender takes advantage of the error-free connection to refill the receiver playout buffer. In the fixed scheduling case, instead, the sending rate is constant and always equal to 192 kb/s. We appreciate only sporadic spikes in the rate of the received data because of the varying percentage of dropped packets and of the flushing of networks buffers when the channel moves from a bad state to a good state.

More importantly, with the proposed algorithm the packet loss rate (PLR) drops significantly from 3.54% to 0.5%, a reduction by a factor of seven. This means that over the 300 second trace used for the simulations, the two techniques lost 10.5 and 1.48 seconds of audio respectively. Figure 5-5 shows the cumulative sum of the audio samples lost throughout the simulation. We observe that especially during very noisy channel conditions (i.e., when the wireless connection is not reliable), the fixed scheduling

technique loses far more packets than the adaptive algorithm. This can be explained by the sensitivity of the proposed transmission protocol that reacts to changes in network conditions and almost stops delivering packets when the feedback from the client warns of the bad channel behavior. In this case the rate adaptation algorithm is not reacting to a congestion event but to a link failure, so it would be worthless to instruct the sender to change the source rate, instead the benefit comes from reducing the sending rate in terms of transmitted packets per second.

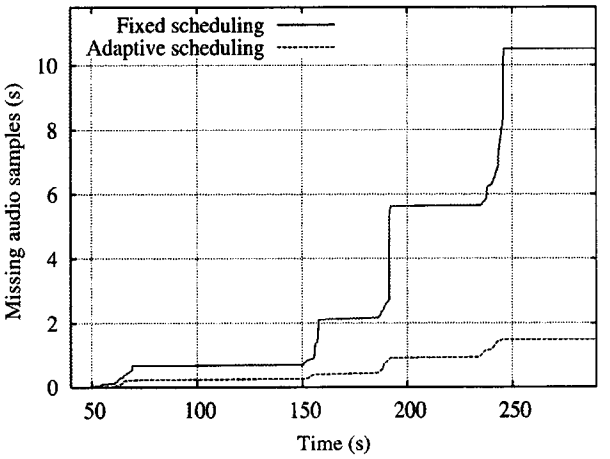


Figure 5-5. Cumulative sum of missing audio seconds during the playout caused by packet losses.

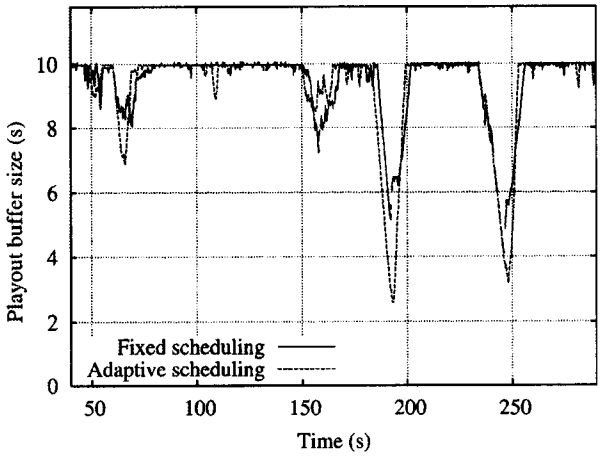


Figure 5-6. Playout buffer size of the two packet scheduling techniques as a function of time.

The playout buffer at the client allows the sender to vary the actual rate of the transmitted data. However, the buffer can accommodate only a certain amount of delay variation depending on its dimension and on the initial pre-buffering period. If this value is overrun, it cannot be guaranteed that the streaming can be continued without interruptions. Hence, it is important to check that the APS algorithm does not have a detrimental effect on the playout buffer. The buffer occupation in the client is shown in Fig. 5-6. In this case an initial pre-buffering period of ten seconds is chosen. The buffer fullness depends on the channel error rate and on the packet delay, so both the transmission scheduling algorithms are affected by the problem of buffer depletion. But, while the fixed scheduling algorithms mainly suffers from the number of lost packets (because its transmission rate is constant), the APS algorithm experiences a reduction of the buffer size also because it delays the transmission of multimedia packets when the channel is bad. The two algorithms, however, present almost the same behavior proving that, if the APS algorithm is able to appropriately adapt the IPG to the channel conditions, not only the PLR is reduced, but also the playout continuity is not damaged by the effect of postponing the transmission of the audio packets with respect to the fixed transmission policy.

4. CONCLUSION

We have investigated the performance of an adaptive packet scheduling algorithm for multimedia streaming in an inter-vehicle network consisting of 802.11b wireless devices. The proposed technique can adapt to the streaming rate by varying the inter-packet transmission interval as a function of the estimated link availability and bandwidth. Network simulation results, based on the network traces collected in our experiments, show that this mechanism is particularly good at reducing the packet loss rate and at providing continuous streaming to an inter-vehicle user with only a limited increase in the packet delay and jitter compared to widely deployed fixed transmission policies.

ACKNOWLEDGMENT

The authors are grateful to Professor Fumitada Itakura of Meijo University and Professors Kazuya Takeda and Nobuo Kawaguchi of Nagoya University for support to this project.

REFERENCES

[1] J.P. Singh, N. Bambos, B. Srinivasan, and D. Clawin, "Wireless LAN performance under varied stress conditions in vehicular traffic scenarios," in Proceedings of IEEE Vehicular Technology Conference (VTC), Vancouver, Canada, September 2002, vol. 2, pp. 743-747.

[2] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.11b for automobile users," in Proceedings of IEEE INFOCOM, Honk Kong, March 2004.

[3] P. Bucciol, E. Masala, N. Kawaguchi, K. Takeda, and J.C. De Martin, "Performance evaluation of H.264 video streaming over inter-vehicular 802.11 ad-hoc networks," in Proceedings of 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communication, Berlin, Germany, September 2005.

[4] D. Wu, Y.T. Hou, W. Zhu, Y-Q. Zhang, and J.M. Peha, "Streaming video over the Internet: Approaches and directions," IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 3, March 2001.

[5] L.-C. Tseng, H.-C. Chuang, C.Y. Huang, and T. Chiang, "A buffer-feedback rate control method for video streaming over mobile communication systems," in Proc. IEEE Intl. Conf. on Wireless Networks Communications and Mobile Computing, Maui, Hawaii, USA, June 2005.

[6] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," IEEE/ACM Transactions on Networking, vol. 7, no. 4, August 1999.

[7] W. Feng and J. Rexford, "Performance evaluation of smoothing algorithms for transmitting prerecorded variable-bit-rate video," IEEE Transactions on Multimedia, vol. 1, no. 3, pp. 301-312, September 1999.

[8] W. Stallings, "Supporting next generation Internet applications today," vol. 1, no. 1, pp. 29-34, Jan-Feb 1999.

[9] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate based congestion control mechanism for real-time streams in the Internet," in Proceedings of INFOCOM, March 1999, pp. 1337–1345.

[10] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP friendly rate control (TFRC): Protocol specification," RFC 3448, January 2003.

[11] S. Floyd and E. Kohler, "TCP friendly rate control (TFRC) for voice: VoIP variant and faster restart," draft-ietf-dccp-tfrc-voip-01.txt, February 2005.

[12] E. Masala, D. Quaglia, and J.C. De Martin, "Variable time-scale streaming for multimedia transmission over IP networks," in Proceedings of 13th European Signal Processing Conference EUSIPCO, Antalya, Turkey, September 2005.

[13] UCB/LBNL/VINT, "Network Simulator - NS - version 2," *URL:* http://www.isi.edu/nsnam/ns, 1997.

Chapter 6

A CONFIGURABLE DISTRIBUTED SPEECH RECOGNITION SYSTEM

Haitian Xu[1], Zheng-Hua Tan[1], Paul Dalsgaard[1], Ralf Mattethat[2], and Børge Lindberg[1]

[1]Speech and Multimedia Communication (SMC), Center for TeleInFrastruktur (CTIF), Aalborg University, Denmark;[2]Technology Institute, Århus, Denmark

Abstract: The growth in wireless communication and mobile devices has supported the development of distributed speech recognition (DSR) technology. During the last decade this has led to the establishment of ETSI-DSR standards and an increased interest in research aimed at systems exploiting DSR. So far, however, DSR-based systems executing on mobile devices are only in their infancy. One of the reasons is the lack of easy-to-use software development packages. This chapter presents a prototype version of a configurable DSR system for the development of speech enabled applications on mobile devices.

The system is implemented on the basis of the ETSI-DSR advanced front-end and the SPHINX IV recognizer. A dedicated protocol is defined for the communication between the DSR client and the recognition server supporting simultaneous access from a number of clients. This makes it possible for different clients to create and configure recognition tasks on the basis of a set of predefined recognition modes.

Key words: distributed speech recognition, noise robustness, application programming interface, fixed-point optimization

1. INTRODUCTION

It is expected that the growth in wireless communication and mobile devices will enable ubiquitous access to a large pool of information resources and services. To make such development successful there is a demand to include ASR as a key component into the user interface. Present mobile devices only have limited memory and CPU capacities which pose

several challenges to ASR. As a result most ASR systems today executing on mobile devices only support low-complexity recognition tasks such as simple name dialling.

The resource-limitation can be partly alleviated by adopting a client-server based DSR architecture[1] as shown in Figure 6-1. In the client, speech signal is recorded and features are extracted from the signal. After the vector quantisation (VQ) and channel coding, the features are transmitted to the server side through the network. In the server, the time-consuming recognition decoding is conducted on the basis of the decoded features. With such architecture the resource limited client becomes independent of the recognition tasks and therefore enables the implementation of complex recognition tasks - e.g. large vocabulary continuous speech recognition.

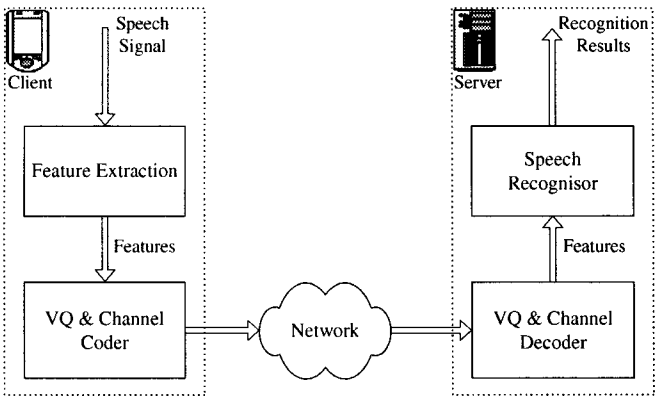


Figure 6-1. The DSR architecture.

Another challenge in deploying speech recognition in mobile devices is the continually changing acoustic and network environment. On the one hand, the speech signal is often corrupted by both the additive and convolutional noise in the acoustic environment resulting in the degradation of the recognitin performance. The deployment of noise-robust signal processing techniques is generally required and normally leads to high computational complexity. The introduction of noise robustness techniques in the DSR architecture may be achieved in the client e.g. by feature enhancement. At the server side, this can be accomplished e.g. by compensating acoustic models or modified decoding strategies. On the other hand, the changing of the network environment may also deteriorate the ASR performance by introducing the transmission errors. Error concealment techniques need to be adopted in the server side to mitigate this effect.

For more than a decade research in the DSR area has led to the establishment of a number of ETSI-DSR standards. The first standard[2] for

the cepstral features was published in the year 2000 with the aim of handling the degradation of ASR over mobile channels caused by both lossy speech coding and transmission errors and enabling interoperability over mobile networks. Currently, one of the most well known standards – the ETSI-DSR advanced front-end (AFE)[3] – further includes the client-side and server-side technqiues providing the DSR system with excellent capabilities for the noise-robustness and error concealment. However, even given these DSR standards, it is infrequent to find real-life DSR implementations executing in standard mobile devices, thus manifesting a barrier for applying the DSR technology in speech driven applications.

This chapter presents a configurable DSR system that has recently been developed at Aalborg University. The AFE is integrated as part of the client and the SPHINX IV[4] is employed as the back-end recognizer. The AFE as used in the system is modified by optimising some time consuming parts of the FFT algorithm. The system is able to support flexible communication to a number of independent user devices each with different requirements to the complexity of the recognition task (e.g. different vocabularies, grammars, etc).

This chapter gives a detailed introduction to this system including its architecture, design considerations and evaluation results. The remainder of this chapter is organised as follows. Section 2 presents the system architecture; section 3 describes a number of considerations for system design and implementation and the system evaluation results are provided in section 4. A case study for a real system is introduced in section 5, and the conclusions are given in section 6.

2. SYSTEM ARCHITECTURE

2.1 System Architecture

As illustrated in Figure 6-2, an embedded *Recorder* in the client simply collects speech signals using a pre-defined sampling rate. The optimised *AFE client-side* module[3] enhances input speech data and generates voice activity detection (VAD) information which together with the set of cepstral features are encoded sequentially and packed into speech packages for network transmission.

At the server side the received speech packages are processed by the *AFE server-side* module. Firstly - on the detection of transmission errors - error concealment is conducted for feature reconstruction. Secondly, the error-corrected speech packages are decoded into a set of cepstral features and VAD information. Subsequently, the cepstral features are processed by the

SPHINX speech recognizer. The recognizer presents its result (either the best or N-best results) at the utterance end – detected by the VAD information - and transmits back to the *Result Listener* of the client. To increase system usability and flexibility, three typical recognition modes are represented, namely: *Isolated word recognition, Grammar based recognition* and *Large vocabulary recognition*. Each is defined by a set of prototype files at the server side. The choice is done at system initialisation, and specific settings can be changed at any time. The setting may be different across a group of end-users.

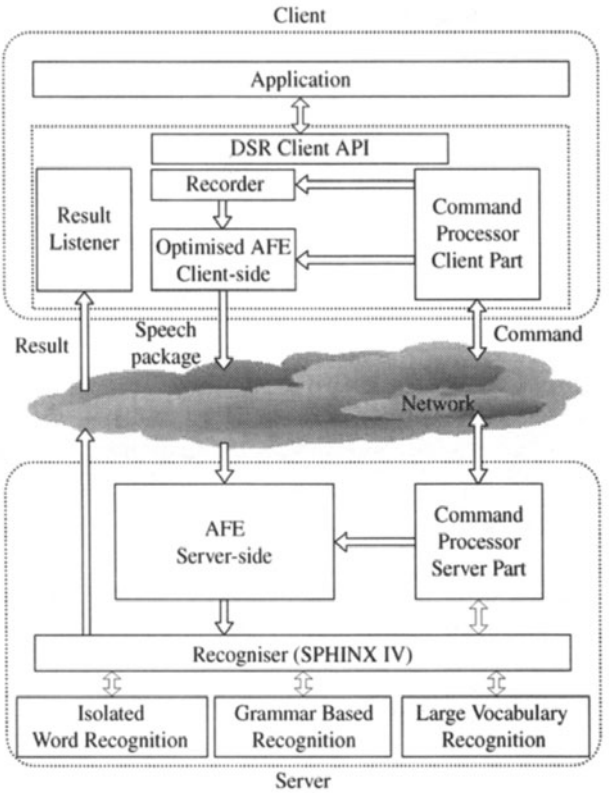


Figure 6-2. The system architecture.

A *Command Processor* is implemented at both the client and server side to support the interchange of configuration commands. Potential commands include control commands to start or stop recognition, choice of recognition mode, commands providing feedback information from the server to a client (e.g. success or failure of any user request), etc.

2.2 Data Streams and Network Load

In the current implementation, two network connections are established for each client accessing the system. The first is the data channel for transmitting speech packages and recognition results, and the second is the control channel for the transmission of control commands. Both connections are socket-based arrangements.

During the recognition process the majority of network load is on the data channel. In full consistency with the DSR standard[3] using an 8.0 kHz sampling rate, the load on the data channel is about 5.6 kbps. The control channel load is small and negligible as compared to the data channel and varies only with the user control settings. With this limited overall bandwidth requirements, the system can be successfully operated over almost all kinds of networks.

2.3 Client Interactions with Potential Applications

The client is implemented in C/C++ for efficiency and portability across different devices and operating systems.

The client is encapsulated into a single dynamical link library (DLL) which supports a series of simple-to-use Application Programming Interfaces (API). For some of them (e.g. functions for acquiring results) different manners are offered. The client has so far been used together with applications written in C, C++, Java or C#, demonstrating its compatibility.

Currently available API functions are listed in Figure 6-3 where they are separated into six clusters covering the following overall functionalities:

- *Initialisation and release functions*: to allocate or de-allocate resources in the client;
- *Network functions*: to connect or disconnect a client and the server. The *Connect* function takes three parameters specifying the server IP address, the port number and the recognition mode;
- *Grammar control functions*: to configure the active vocabularies or rule grammars in a Java Speech Grammar Format[9] for the isolated word and grammar-based modes, respectively;
- *Recognition control functions*: to start or stop the recognizer at the server;
- *Results related functions*: to acquire results. Both synchronous and asynchronous manners are provided for the application. With the synchronous manner, the application inserts a call back function through *SetCallBackFunction* by which it receives a "callback" notification when the result is ready. With the asynchronous manner, the *GetResult* function is called by the application and only returns when the result is ready.

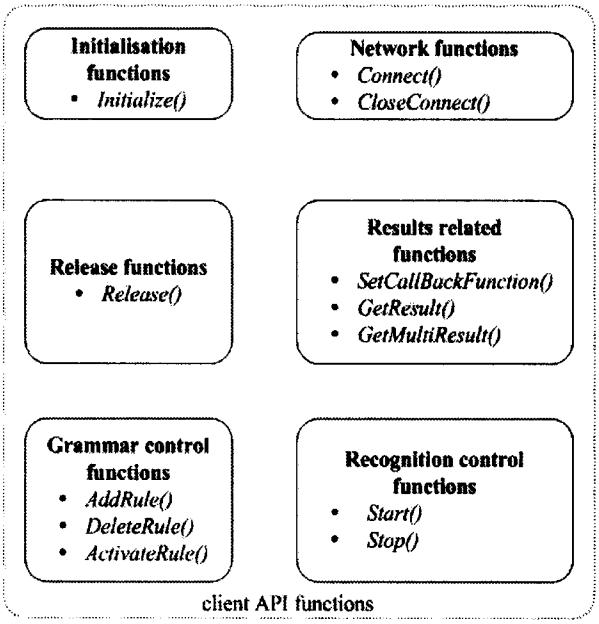


Figure 6-3. Client API functions.

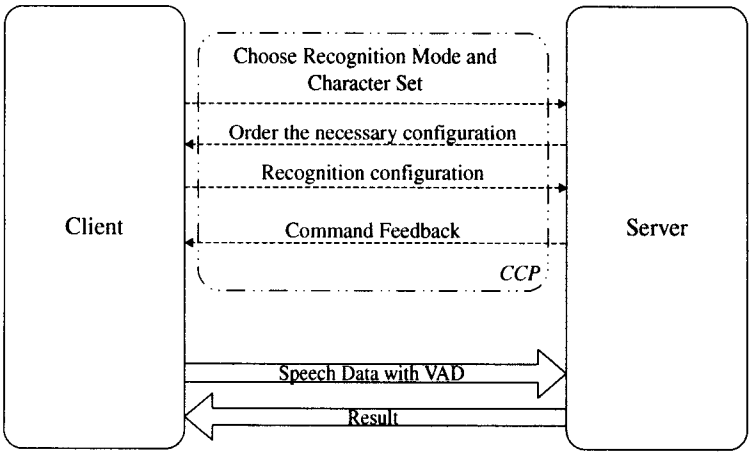


Figure 6-4. CCP in client-server communication.

3. SYSTEM IMPLEMENTATION CRITERIA

The design and implementation of the system require a number of special overall considerations to be accounted for at both the client and the server side.

3.1 Command Control Protocol (CCP) and Configurability

Proper communication between the client and the server is supported by the *Command processor* together with a simple communication protocol CCP as illustrated in Figure 6-4. The CCP consists of a number of control commands that submit the requirements from the client and acquire the feedback from the server. Specifically the client may choose the character set and recognition mode, change recognizer configuration such as grammars, control recognition start and stop etc. The client may be notified by the server about the success or failure of server actions for each client control command. Each control command is acknowledged to ensure correct transmission of commands over the command channel.

Through the means of the CCP the client can flexibly configure the system functionality.

3.2 Noise Robustness and Error Concealment

The acoustic noise and network transmission errors may severely affect the performance of the DSR system. In this work, they are partially conquered by employing the Aurora AFE. The AFE client part performs the noise reduction by using a two-stage Mel-warped Wiener filter noise reduction algorithm. Then the signal to noise ratio (SNR)-dependent waveform processing is applied to emphasize the high SNR portions of waveform whereas de-emphasizing the low SNR portions. The blind equalisation is finally used to mitigate the convolutional noise.

In the server, the transmission errors are detected and mitigated by an error mitigation module.

3.3 Client Efficiency

Today the CPU in most mobile devices is only able to conduct floating point arithmetic by automatically converting it into fixed point calculations. This causes a dramatic drop in computational efficiency. The system presented in this chapter integrates a fixed-point AFE implementation. This

executes several times faster than the floating-point AFE. It is further optimised for one of the most popular CPUs used in PDAs, the Intel® XScale[5] by substituting high-level language instructions with CPU-dependent ASM instructions within the most time-critical part of the FFT module.

Finally, multi-thread programming[6] is employed with the goal of utilizing the CPU time optimally.

3.4 Choice and Optimisation of Speech Recognizer

The speech recognizer is the primary module in the server controlling not only the recognition performance but also the flexibility and usability of the system. There are several advantages given by the choice of the SPHINX IV recognizer.

- SPHINX IV is based on an object-oriented design which makes its integration with other DSR modules simple;
- SPHINX IV supports a wide range of recognition tasks rendering it dynamically configurable;
- SPHINX IV is computationally efficient and has shown good recognition performance[4].

However, during system development it is observed that the loading time for acoustic and language models for a large vocabulary speech recognition task is rather long. This has resulted in the optimisation of part of the Java source code in the recognizer by substituting them with more efficient C code.

3.5 Support of Character Sets

Given the fact that the ANSI character set is not always supported in mobile devices, both the ANSI and the UNICODE character sets are supported in communicating commands and results between the client and the server.

3.6 Choice of Recognition Mode

Each of the modes includes a set of pre-trained acoustic models, a SPHINX recognizer configuration file (in XML format) and a vocabulary. Additionally a language model should be contained for the *Large vocabulary recognition* mode. Wordlists (active vocabularies) and grammars are dynamically configured by the application through the APIs.

4. EVALUATIONS

The DSR system described in this chapter is evaluated with respect to its recognition accuracy and time efficiency.

4.1 Recognition Performance

Four tests have been conducted each on a recognizer with the cepstral features calculated either by a floating point or by a fixed point AFE - with or without a vector quantizer (VQ). The first two tests use the HTK recognizer[8]. The word models used with these tests are trained using the HTK training software. The latter two tests deploy the SPHINX recognizer. The word models are trained by the SPHINX training tools.

The speech data used for all tests are from the English connected digits recognition task in the Aurora 2 database[7]. Each digit is modelled by 16 HMM states each with three Gaussian mixtures. The acoustic models are trained using the "Multi-condition training" settings in[7] where clean speech and noise data of test Set A are added with SNR ranging from 20dB to 0dB. The training data thus includes four types of noise ("Subway", "Babble", "Car" and "Exhibition"). The speech features are the normally used 39-dimensional MFCC vector.

The average word accuracies for the four test sessions across the four set of test data are shown in Table 6-1.

Table 6-1. Test set A word accuracy (%).

	Subway	Babble	Car	Exhibition	Average
HTK (Floating point)	91.64	90.30	93.77	91.46	91.79
HTK (Floating point, VQ)	91.37	90.19	93.48	91.54	91.65
SPHINX (Floating point, VQ)	89.71	90.19	91.45	90.19	90.38
SPHINX (Fixed point, VQ)	89.73	90.18	91.44	90.17	90.38

It is observed that the vector quantised AFE features, only cause the recognition accuracy to drop slightly and rather uniformly across the four types of noise data. With the current setting of the two recognizers, the SPHINX IV shows lower averaged word accuracy as compared to the floating point HTK recognizer. It is noted that the fixed-point SPHINX

recognizer gives results that are very close to those resulting from the floating point AFE.

4.2 Client Resource Consumptions

Tests have been conducted on PC or PDA devices running either Windows or Windows CE (Pocket PC 2002 or 2003) using wired or WiFi network connections.

The size of the client DLL library file is limited to only about 74Kbytes, and the highest memory consumption at run-time is less than 29Kbytes.

The optimisation described in section 3.3 is evaluated on a H5550 IPAQ with a 400MHz XScale CPU and 128 MB memory. The data used is an 11-second test sound clip sampled at 8 KHz. The overall test results are provided in Table 6-2 demonstrating large computational reduction in execution time when replacing the floating point AFE with the fixed point algorithm in combination with FFT optimisation.

Table 6-2. Real-time efficiency in using different realizations of the AFE.

Algorithm	Floating Point AFE	Fixed Point AFE	Fixed Point AFE + FFT optimisation
x Real-time	3.98	0.82	0.69

5. A CASE STUDY

To provide general ideas about the developing of potential speech-driven applications based on the implemented DSR system, this section presents an example application under development - a spoken query answering (SQA) system[10].

To establish the information and service accessibility in mobile devices with limited resources, the system applies DSR and knowledge-based Information Retrieval (IR) processing for spoken query answering (SQA) as shown in Figure 6-5. Specifically, a Graphical User Interface (GUI) is implemented in the client based on the DSR API. This makes it possible for the user to input information queries by speaking directly to the mobile devices. The query text is obtained from the input speech by the DSR and transmitted to the IR server for the information retrieval. The retrieved documents are displayed in the client GUI. The system can currently answer queries and questions in Danish within the chosen soccer test domain.

A grammar-based recognition mode is built on the DSR server. The acoustic model used is the tri-phone model. With the tools from the

SphinxTrain[4], the model training is achieved based on the 91 hours of speech from the SpeechDat-2 database[11], which contains telephony speech sampled at 8.0 kHz. A series of rule-grammars are defined in the DSR client allowing asking questions about players, teams and matches. During the system initialisation, the grammars are added to the DSR server by the corresponding client API functions.

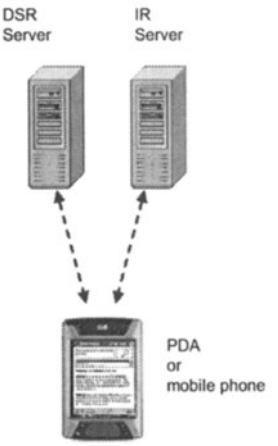


Figure 6-5. The architecture of the speech-driven IR system.

During the development of the SQA system, we experienced the easiness and efficiency of integrating the speech part by using the introduced DSR system. The client GUI programming is based on the C#, and the special considerations about the flexibility and the compatibility of the DSR APIs to different programming languages make the programming effortless.

6. CONCLUSION

This chapter introduces a DSR-based system that is developed on the basis of the AFE and the SPHINX IV speech recognizer. The system is designed for being configurable and facilitating simultaneous access from a number of clients each with its own requirements to the recognition task. A system communication protocol is designed to control the interaction between the client and the server. The recognizer supports multiple recognition modes covering isolated word recognition tasks, grammar based recognition tasks, and large vocabulary continuous speech recognition tasks.

The system has been tested and shows high performances both in respect to real-time efficiency and the recognition accuracy.

Finally, an SQA system is introduced as a case study to give general ideas about integrating the DSR system into real-life applications and demonstrates the easiness and efficiency.

ACKNOWLEDGEMENT

This project is supported by the CNTK (Centre for Network and Service Convergence) project which is funded partly by the Danish Ministry of Research, Technology and Development and partly by the industrial partners. It includes participations from Danish telecommunication companies and two Danish technical universities. Furthermore, the authors wish to thank Dr. David Pearce, Motorola Corporation for providing the help for the fixed-point AFE client implementation, and also the POSH project for its support of the SQA system.

REFERENCES

[1] Z. -H. Tan, P. Dalsgaard and B. Lindberg, "Automatic speech recognition over error-prone wireless networks," Speech Communication, 2005.

[2] ETSI draft standard doc. Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms, ETSI ES 202 108 V1.1.2 (2000-04), April 2000.

[3] 3GPP TS 26.243: "ANSI-C code for the Fixed-Point Distributed Speech Recognition Extended. Advanced Front-end", December, 2004.

[4] W. Walker, P. Lamere, and P. Kwok et al. "SHPHINX-4: A Flexible Open Source Framework for Speech Recognition", Technical report TR-2004-139, Sun corporation, USA, 2004.

[5] Intel® XScale technology overview: http://www.intel.com/design/intelxscale/index.htm

[6] J. Richter, *Programming applications for Microsoft Windows*, 4th Edition, Microsoft Press 1999, USA.

[7] H. G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions", ISCA ITRW ASR2000 (Automatic Speech Recognition: Challenges for the Next Millennium), Paris, France, September 18 - 20, 2000.

[8] S. Young. "HTK: Hidden Markov Model Toolkit V1.5". Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington DC, Dec. 1993.

[9] "Grammar Format Specification", Technical documentation, Sun Microsystems, Inc, October, 1998.

[10] T. Brøndsted, H. L. Larsen, L. B. Larsen, B. Lindberg, D. Ortiz-Arroyo, Z .-H. Tan and H. Xu, "Mobile Information Access with Spoken Query Answering", The Proc. of ISCA ITRW - the ASIDE2005/COST278 Final Workshop, Nov. 2005, Aalborg, Denmark.

[11] B. Lindberg "Speechdat, Danish FDB 4000 speaker database for the fixed telephone network", pp. 1–98, March 1999.

Chapter 7

EMBEDDED MOBILE PHONE DIGIT-RECOGNITION

Christophe Lévy, Georges Linarès, Pascal Nocera, and Jean-François Bonastre

Laboratoire Informatique Avignon, 339 chemin des Meinajaries, BP 1228, 84911 AVIGNON, France

Abstract: Speech recognition applications are known to require substantial amount of resources in terms of training data, memory and computing power. However, the targeted context of this work - embedded mobile phone speech recognition systems - only authorizes few KB of memory, few MIPS and usually a small amount of training data. In order to meet the resource constraints, an approach based on an HMM system using a GMM-based state-independent acoustic modeling is proposed in this paper. A transformation is computed and applied to the global GMM in order to obtain each of the HMM state-dependent probability density functions. This strategy aims at storing only the transformation function parameters for each state and enables to decrease the amount of computing power needed for the likelihood computation. The proposed approach is evaluated with a digit recognition task using the French corpus BDSON. Our method allows a Digit Error Rate (DER) of 2.1%, when the system respects the resource constraints. Compared to a standard HMM with comparable resources, our approach achieved a relative DER decrease of about 52%.

Key words: Embedded speech recognition, isolated digit-recognition, GMM-based approach

1. INTRODUCTION

The amount of services provided by the latest generation of mobile phones has increased significantly compared to oldest models. Nowadays,

phones offer new kinds of feature such as organizer, phone book, e-mail/fax, or games. At the same time, the size of mobile phones has reduced significantly. Both evolutions raise an important issue: "How to use all the services of mobile phone without a large keyboard in platforms like the communication inside a moving car?" Voice-based human-to-computer interfaces (HCI) supply a friendly solution to this problem but require embedding a speech recognizer into the mobile phone.

Since the last decade, the performance of Automatic Speech Recognition (ASR) systems has improved and now allows building efficient voice human-to-computer interfaces. Moreover, even if scientific progress has been made, the gain (in performance) remains linked to computing resources: a last-generation computer with a lot of memory is generally required. Consequently, the main problem to embed ASR into a mobile phone is the low level of resource available, which usually consists of a 50/100 MHz processor, a 50/100 MHz DSP, and less than 100KB of memory.

State-of-the-art speech recognition systems are mainly related to statistical methods like the Hidden Markov Model. These stochastic ASR systems obtain good results. Nevertheless, to get these performances a large training data is required. Moreover, the training data should be as close as possible to the targeted application. For embedded mobile phone speech processing, few speech corpora are available (most of the collected speech material is not directly recorded into a mobile phone, which adds coding and transmission problems). In order to cope with this problem, the acoustic models are generally trained using large corpora recorded in different conditions before being adapted to the targeted context, using the limited amount of data available.

The mobile phone context involves large environment variability as clients use their mobile phones in multiple locations (office, car, street, etc.). In order to improve the speech robustness in these adverse conditions, large acoustic models (trained with enough data) and/or acoustic-model adaptation are required. Nevertheless, mobile phone resource constraints emphasize the need for new, less costly, solutions.

In this chapter, we focus primarily on the memory constraints, even if the solution proposed allows a significant gain in computational cost and reduces the requirement for training data. Our approach consists in modeling the acoustic space with a unique GMM, which is derived in order to obtain each HMM-state probability density function by applying a simple transformation (*cf. Figure 7-1*). This approach uses a phone-based ASR system with a dynamic lexicon. It allows recognition of various vocabularies such as digit recognition, name dialing, voice command, and others.

In this context, only the transformation parameters need to be stored for a given state. This approach also allows a gain in computational cost as a part of the likelihood computation is shared between the states and/or between components. Our approach, firstly proposed in Lévy *et al.* (2005), is technically close to Young's (1992), Park and Ko's (2004) or Bonastre *et al.* (2003). The main difference concerns the training of the acoustic model: as a GMM or as an HMM with tied components. Other approaches for embedding classical HMM recognizers in mobile phones are also present in the literature, like in Astrov *et al.* (2003) especially focused on the reduction of feature vector dimension.

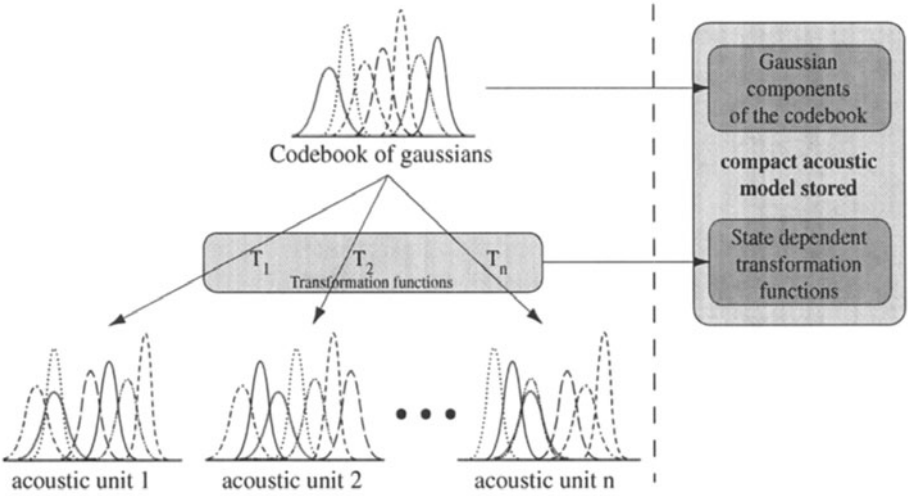


Figure 7-1. Overview of the proposed approach.

In section 2, we present the corpora used for the experiments. Then, a baseline HMM system is presented in section 3. Section 4 describes our proposed approach. Section 5 shows some experimental results and some ongoing work and conclusions are discussed in sections 6.

2. CORPORA

In this study, the database BDSON is used for evaluating our approach. A second database, BREF, is used only to train the state-independent GMM model. Both are French databases collected in clean acoustic environments.

2.1 BREF

BREF (Lamel *et al.* 1991) is a large read-speech corpus composed of sentences selected from the French newspaper *"Le Monde"*. This corpus contains about 100 hours of speech material from 120 speakers.

This corpus is used for training the baseline HMM system and to estimate the state-independent GMM (the models are trained using BREF and then adapted to BDSON as explained in the next paragraph).

2.2 BDSON

BDSON (Carré *et al.* 1984) includes speech material (in French) recorded from 30 speakers (15 male and 15 female speakers). We divided BDSON into two parts:

- One for the application-context adaptation (ADAPT_SET): it includes 700 digits pronounced by 7 speakers (4 male and 3 female speakers). This set is used to adapt the baseline HMM and the state-independent GMM to the application context, here the BDSON corpus. This is done once, and the rest of the paper will refer to these adapted models, both for the baseline HMM and for the state-independent GMM;
- Another one for evaluating the performance of the systems (TEST_SET): composed of 2300 utterances of digits pronounced by 23 speakers (11 male and 12 female speakers).

The speakers of ADAPT_SET are different from the speakers of TEST_SET (and are also different from the BREF speakers). The systems are speaker independent and no speaker adaptation is applied.

The objective in this paper is isolated word recognition embedded into a mobile phone. The system should be able to recognize digits, voice commands and proper names. A dynamic lexicon is used, based on the phonemic description of words. Due to database considerations, the performance is evaluated thanks to digit recognition. The digits are considered as words (*i.e.* no specific adaptation of the system is done, such as reduction in the number of phoneme models). The performance measure is denoted Digit Error Rate (DER) in this paper.

3. BASELINE HMM SYSTEM

In order to evaluate the performance of our approach depending on the level of memory resource targeted, a set of classical HMMs of various sizes

is firstly trained and tested. These first experiments provide the performance of the baseline system for each model size.

The training process is composed of two successive stages. The models are trained firstly using the BREF corpus. Then, the second stage consists in adapting these models (weight, mean and variance) with ADAPT_SET using the MAP approach (Gauvain *et al.* 1994).

The models are composed of 38 phonemic models and 108 emitting states (context-independent models). The number of Gaussian components per state varies between 128 for the largest model to 2 for the smallest one. 39 PLP coefficients per frame (13 static and first and second derivative) are used for the largest model when only 13 (static PLP) are used for the others.

The largest model complexity is about 1 million parameters while the smallest model is using only 5800 parameters.

Table 7-1 shows a DER about 0.96% when using the full-size baseline system (128 Gaussian components per state and 39 coefficients for the acoustic vectors). The DER increases to 4.43% and 4.96% (in absolute DER) when the acoustic model size meets the targeted context (model size is decreased by a factor between 100 and 200).

Table 7-1. DER for baseline HMM based on number of parameters in the model (2300 tests). Two last lines represent the models with acceptable size for the mobile phone applications.

	DER	Number of parameters
128g. (39 coef.)	0.96%	1092k
4g. (13 coef.)	4.43%	11k
2g. (13 coef.)	4.96%	6k

4. PROPOSED GMM-BASED APPROACH

Our approach consists in modeling the acoustic space using a unique GMM and then in deriving the state dependent Probability Density Functions (PDF) from it. The basic transformation function to obtain the state-dependent GMM is an adaptation of the weight parameters (*cf.* 4.1) followed by a *NBest* component selection: only the weights of the winning components are memorized for a given state-dependent GMM. An optional phase is also proposed, where the general GMM model is adapted firstly using the same transformation function for all the components (*cf.* 4.2).

4.1 Weight Re-Estimation (WRE)

Two criteria are used in order to adapt the GMM Gaussian weight:

- Maximum Likelihood Estimation (MLE),

- Maximum Mutual Information Estimation (MMIE).

After performing this weight adaptation, only the *NBest* Gaussian components are stored in order to decrease the memory occupation. Furthermore, the likelihood for each component of the global GMM is computed only once and then all the state likelihoods are computed easily thanks to a simple weighted combination of the individual-component likelihoods.

4.1.1 MLE

This approach consists in estimating the state-dependent weight vector from the initial GMM and an HMM-based frame alignment. Then, each state is represented by the state-independent GMM component set and by its specific weight vector.

The Gaussian weights w_i are re-estimated using a Maximum Likelihood Estimation criterion defined by:

$$w_i' = \frac{w_i * L(X / g_i)}{\sum_{g_j=1}^{nb_g} w_j * L(X / g_j)} \tag{1}$$

where w_i is the *a priori* weight of the i^{th} Gaussian component and $L(X / g_i)$ is the likelihood for the state-related frames (X) for a given component (g_i) of the global GMM.

4.1.2 MMIE

HMM training using Maximum Mutual Information Estimation has been studied largely in the last years (Bahl *et al.* (1986)). This discriminative approach consists in estimating the model parameter λ, which maximize the objective function F_λ:

$$F_\lambda = \sum_{r=1}^{R} \log\left(\frac{P_\lambda(O_r / M_{w_r}) * P(W_r)}{\sum_{\hat{W}} P_\lambda(O_r / M_{\hat{w}}) * P(\hat{W})} \right) \tag{2}$$

where O_r represents the observation sequence, W_r the correct transcription, $\hat{W}$ the wrong transcriptions, M_{W_r} and $M_{\hat{W}}$, the models associated with the correct and wrong transcriptions.

Several papers report significant improvement of system performance when large training corpora are available. Moreover, the parameter-estimation computational-cost remains very expensive in spite of several fast approximates (based on *NBest* decoding, Gaussian selection, word or phone lattices …).

Woodland and Povey (2000) propose a weight estimation rule based on the maximization of the cost function:

$$F_W(\lambda) = \sum_{m=1}^{M} \left[\gamma_{jm}^{num} * \log(\hat{c}_{jm}) - \frac{\gamma_{jm}^{den}}{c_{jm}} * \hat{c}_{jm} \right] \tag{3}$$

where γ_{jm} is the sum over the time of the probabilities to be in state j and mixture component m.

The maximum of $F_W(\lambda)$ can be reached iteratively by finding the optimal of one weight holding the others. Then, maximization of (3) can be performed by optimising each term of the sum, which is a convex function. We obtain:

$$\hat{c}_{jm} = \arg\max_{c} \left(\gamma_{jm}^{num} * \log(c) - \frac{\gamma_{jm}^{den}}{c_{jm}} c \right) = \frac{\gamma_{jm}^{num}}{\gamma_{jm}^{den}} c_{jm} \tag{4}$$

This weight estimation is used for ULT+MMIE approach. For the weight re-estimation in the specific case of full Gaussian sharing (without ULT), we propose a fast updating function. The formulas are rewritten focusing on the Gaussian contributions in γ probabilities. Denoting $L(X/G_{jm})$ and $L(X/Si)$ the likelihood of X knowing respectively Gaussian component G_{jm} and the state S_i, Ω^k the set of frames emitted by the state k, (4) can be develop as follows:

$$\hat{c}_{jm} = c_{jm} * \frac{\gamma_{jm}^{j}}{\sum_{k} \gamma_{jm}^{k}} \tag{5}$$

$$\gamma_{jm}^{k} = \sum_{X \in \Omega^k} \frac{L(X/S_j)}{\sum_{i} L(X/S_i)} * \frac{c_{jm} * L(X/G_{jm})}{L(X/S_j)} \tag{6}$$

$$\gamma_{jm}^{k} = \sum_{X \in \Omega^k} \frac{c_{jm} * L(X/G_{jm})}{\sum_i L(X/S_i)} \tag{7}$$

It is worth noting that all Gaussian components are shared and thus equation (7) can be written:

$$\gamma_{jm}^{k} = \sum_{X \in \Omega^k} c_{jm} * \frac{L(X/G_{km})}{L(X/S_k) + \sum_{i \neq k} L(X/S_i)} \tag{8}$$

and

$$\frac{\gamma_{jm}^{num}}{\gamma_{jm}^{den}} = \frac{\sum_{X \in \Omega^k} \frac{L(X/G_{jm})}{L(X/S_j) + \sum_{i \neq k} L(X/S_i)}}{\sum_l \sum_{X \in \Omega^l} \frac{L(X/G_{lm})}{L(X/S_l) + \sum_{i \neq l} L(X/S_i)}} \tag{9}$$

which can be approximated by:

$$\frac{\gamma_{jm}^{num}}{\gamma_{jm}^{den}} \approx \frac{c_{jm}}{\sum_k c_{km}} \tag{10}$$

We obtain finally the weight updating function:

$$\hat{c}_{jm} \approx c_{jm} * \frac{c_{jm}}{\sum_k c_{km}}$$

This result allows discriminative weight re-estimation without any additional training stage by using only the MLE original weights. Experiments show that results obtained (using this fast MMIE weight re-estimation) are closed to ones obtained by standard MMIE updating functions.

4.2 Unique Linear Transformation (ULT)

The LIAMAP method presented in Matrouf *et al.* 2003 allows to adapt globally the initial GMM for a given state, using a unique and simple transformation. This transformation (applied both on the mean and the variance) is a linear adaptation:

$$\mu_{StateGMM} = \alpha * \mu_{GlobalGMM} + \beta$$

$$\sigma_{StateGMM} = \alpha^2 * \sigma_{GlobalGMM}$$

where α (which is common for $\mu_{StateGMM}$ and $\sigma_{StateGMM}$) and β are given as follows. This adaptation (as illustrated by *Figure 7-2*) corresponds to the estimation of a linear transformation between two Gaussians obtained by:

1. Merging the Gaussian components of the initial GMM. The final Gaussian is defined by μ and Σ, respectively the mean and the covariance matrix.
2. Adapting the Gaussian components of the GMM using state-specific data (using MAP) and then merging adapted Gaussians to obtain a unique Gaussian defined by $\tilde{\mu}$ and $\tilde{\Sigma}$.
3. Computing α and β as the parameters of a linear adaptation between the Gaussian (μ, Σ) and the Gaussian $(\tilde{\mu}, \tilde{\Sigma})$.

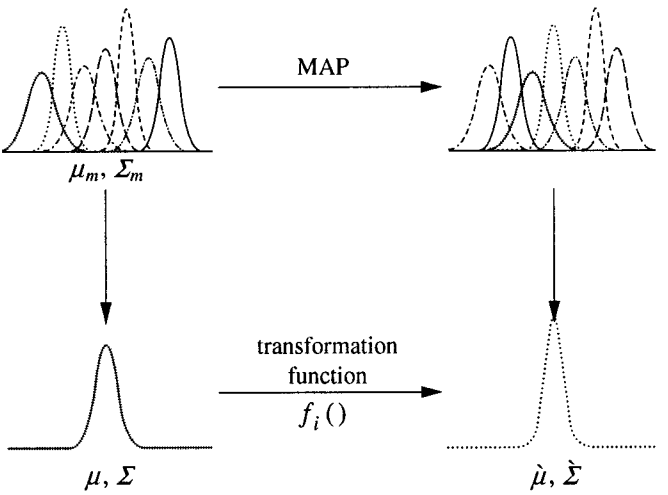


Figure 7-2. LIAMAP: Estimation method of a unique linear transformation for all Gaussians of a codebook.

Each final Gaussian component (defined by its mean μ'_m and its covariance matrix Σ'_m) is computed as follows:

$$\mu'_m = \tilde{\Sigma}^{1/2}\Sigma^{-1/2}(\mu_m - \mu) + \tilde{\mu} \tag{11}$$

$$\Sigma' = \tilde{\Sigma}\Sigma^{-1}\Sigma_m \tag{12}$$

Equation (11) could be expanded as:

$$\mu'_m = \tilde{\Sigma}^{1/2}\Sigma^{-1/2}\mu_m - \tilde{\Sigma}^{1/2}\Sigma^{-1/2}\mu + \tilde{\mu} \tag{13}$$

Let us use

$$\alpha = \tilde{\Sigma}^{1/2}\Sigma^{-1/2} \tag{14}$$

and

$$\beta = -\tilde{\Sigma}^{1/2}\Sigma^{-1/2}\mu + \tilde{\mu} \tag{15}$$

equations (11) and (12) become:

$$\mu'_m = \alpha\mu_m + \beta \tag{16}$$

and

$$\Sigma'_m = \alpha^2\Sigma_m \tag{17}$$

Equations (16) and (17) correspond to a linear adaptation function defined only by the vectors α and β (the transformation is shared by all the Gaussian components of the GMM).

In our context, ULT is used as a first step before the weight adaptation; the following steps are the same as the WRE approach[3]. Figure 7-3 presents the complete process.

During the testing stage, this approach requires to apply the transformation state by state before computing the corresponding likelihood. It implies also to re-compute the component likelihoods for each state.

[3] In this case, the WRE likelihoods in equation (1) are computed using the ULT-transformed components.

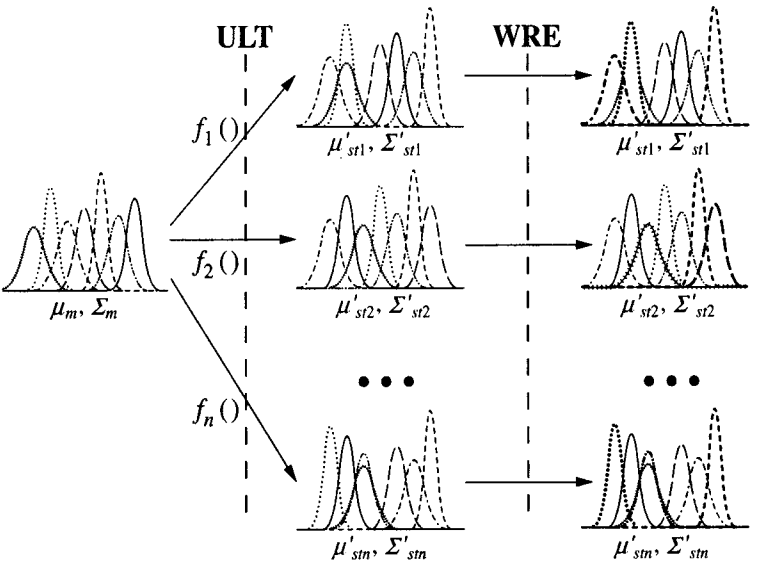


Figure 7-3. State-dependent transformation by applying ULT followed by WRE.

4.3 Evaluation of Memory Occupation

For each approach - HMM baseline system, WRE and WRE+ULT - we estimate the acoustic model sizes (in terms of number of parameters) and select the number of Gaussian components for each system in order to have the same number of parameters in the models.

For the largest models, the number of Gaussian components is limited to four components per state for the baseline system (with an overall total of 432 components), to 365 components for the WRE method and, finally, to 257 components for the ULT+WRE approach. For the smallest models, the number of Gaussian components becomes 2 (per state) for the baseline system, 141 for the WRE approach and 35 for the ULT+WRE (for the global GMM).

5. RESULTS

Experiments on isolated-digit recognition task were carried out on the BDSON corpus in order to evaluate the performance of our approach. For comparison, the same experiment was carried out using the baseline system with the same memory occupation.

For all experiments with approach under consideration (WRE/MLE, WRE/MMIE, ULT+WRE/MLE and ULT+WRE/MMIE), the *NBest* parameter is dynamically set, for each state, by selecting Gaussian components showing maximum weights until the total of selected component weights remains higher than an *a priori* fixed threshold.

Table 7-2. DER for baseline HMM, WRE/MLE, WRE/MMIE, ULT+WRE/MLE and ULT+WRE/MMIE (2300 tests).

Model Size	HMM	WRE/MLE	WRE/MMIE	ULT + WRE/MLE	ULT + WRE/MMIE
6k	4.96%	4.17%	4.52%	3.39%	3.22%
11k	4.43%	3.09%	2.09%	3.00%	2.70%

Regarding *Table 7-2*, the DER is decreased significantly in all the cases when our approach is used: with higher memory constraint the DER decreases from 4.43% to 2.09% (with WRE/MMIE) which represent a relative gain around 52%.

With *6k* models, the best results are obtained using ULT+WRE/MMIE method (DER decreases from 4.96% to 3.22%). Nevertheless, these results are closed to ULT+WRE/MLE ones.

Increasing the model complexity, the performance improvement is also significant. This is especially observed using MMIE weight re-estimation (DER decreases from 4.43% to 2.09%). ULT method obtains relative DER improvements similar to those obtained using *6k* models, both with MMIE and MLE weight estimation.

6. CONCLUSION AND PERSPECTIVES

In this paper, a viable solution is proposed to embed automatic speech recognition into a mobile phone. The technique proposed is based on an HMM with a global, state-independent, GMM modeling of the acoustic space and a set of transformation functions able to adapt this GMM to obtain each state-dependent probability density function. This approach drastically reduces the memory size of the models and the computation time needed to compute the likelihoods (even if, in this paper, the focus is mainly put on memory occupation).

Two different techniques are discussed for the transformation functions. The first one, WRE, consists in adapting the state-independent GMM weights using MLE or MMIE and selecting and storing the *NBest* component weights only. WRE allows to save memory without reducing the

global GMM size and to reduce the likelihood computation time as the component likelihoods are calculated only once per frame. The second one, ULT+WRE, transforms the mean and variance parameters before applying WRE. It uses a unique linear transformation for all the components. ULT+WRE allows a better modeling of state-dependent PDFs (and requires additional computing time).

The results have shown a large gain in terms of performance (a relative reduction of DER, between 10% and 50%) compared to a classical HMM technique with equivalent model memory size. For example, with the 11K parameter acoustic model, the baseline HMM system reaches a 4.4% DER whereas with our approach the DER ranges between 3.09% (WRE/MLE) and 2.09% (WRE/MMIE). For comparison, the DER HMM-based system with a full-size model – more than 1 million parameters – is around 1%.

For the ULT, other transformations can be used such as Discriminative Linear Transformation, approximation of full ULT by adapting only the mean of Gaussian components, *etc.*

In this work, no adaptation has been performed to the speaker or the environment when both factors are known to improve ASR system performance. The structure of our models permits the adaptation of only the state-independent GMM without changing the state-dependent transformations, which could be particularly interesting. A possible strategy for this adaptation could be to use the test data to adapt the common GMM parameters (or a part of the parameter set) before decoding. Even if few frames are available, only one model has to be adapted instead of one model per state for a classical HMM system. Furthermore, this approach does not require any decoding before the adaptation since all the frames are related to only one state. This characteristic removes the influence of decoding errors during the initial decoding while the additional computing cost remains very low. Lastly, data from a word or a phoneme could be used to adapt non-observed models.

The latter perspective seems very promising, particularly for real cell phone data.

REFERENCES

[1] Astrov, S., Bauer, J.G., and Stan, S., 2003, "High performance speaker and vocabulary independent asr technology for mobile phones," in Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP'2003), Hong Kong, pp. 281–284.

[2] Bahl, L., Brown, P., De Souza, P. and Mercer, R., "Maximum mutual information estimation of hidden Markov model parameters for speech recognition", in Proceedings

of International Conference on Acoustics Speech and Signal Processing (ICASSP'1986), Tokyo, Japan, pp. 49–52

[3] Bonastre, J.F., Morin, P. and Junqua, J.C., "Gaussian Dynamic Warping method applied to Text-Dependent speaker detection and verification," in Proceedings of European Conference on Speech Communication and Technology (Eurospeech'2003), Geneva, Switzerland, pp. 2013–2016.

[4] Carré, R., Descout, R., Eskénazi, M., Mariani, J., and Rossi, M., 1984, "The French language database: defining, planning and recording a large database," in Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP'1984), San Diego, California, USA, pp. 324–327.

[5] Gauvain, J.L., and Lee, C.H., 1994, "Maximum A Posteriori estimation for multivariate gaussian mixture observations of markov chains," in IEEE Transactions on Speech and Audio Processing, vol. 2-2, pp. 291–298.

[6] Lamel, L.F., Gauvain, J.L., and Eskénazi, M., 1991, "BREF, a large vocabulary spoken corpus for French," in Proceedings of European Conference on Speech Communication and Technology (Eurospeech'1991), Genoa, Italy, pp. 505–508.

[7] Lévy, C., Linarès, G., Nocera,, P., and Bonastre, J.F., 2005, "Mobile phone embedded digit-recognition," Biennial on DSP for in-Vehicle and Mobile Systems, Sesimbra, Portugal.

[8] Matrouf, D., Bellot, O., Nocera, P., Linarès, G., and Bonastre, J.F., 2003, "Structural linear model-space transformations for speaker adaptation," in Proceedings of European Conference on Speech Communication and Technology (Eurospeech'2003), Geneva, Switzerland, pp. 1625–1628.

[9] Park, J., and Ko, H., 2004, "Compact acoustic model for embedded implementation," in Proceedings of International Conference on Spoken Language Processing (ICSLP'2004), Jeju Island, Korea, pp. 693–696.

[10] Woodland, P. and Povey, D., "Large Scale Discriminative Training for Speech Recognition", In ISCA ITRW Automatic Speech Recognition: Challenges for the Millennium, pages 7-16, Paris, France, 2000.

[11] Young, S.J., 1992, "The general use of tying in phoneme-based HMM speech recognisers," in Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP'1992), San Francisco, California, USA, pp. 569–572.

Chapter 8

ON THE COMPLEXITY-PERFORMANCE TRADEOFF OF TWO ACTIVE NOISE CONTROL SYSTEMS FOR VEHICLES

Pedro Ramos, Luis Vicente, Roberto Torrubia, Ana López, Ana Salinas, and Enrique Masgrau.
Communication Technologies Group (GTC). Aragón Institute for Engineering Research (I3A). University of Zaragoza. Spain.

Abstract: The aim of this chapter is to show primarily the experimental results achieved in the attenuation of periodic disturbances inside a vehicle with two Active Noise Control algorithms implemented on the TMS320C6701 DSP and to compare the computational complexity of both strategies: (i) Modified FxGAL: Modified filtered-x gradient adaptive lattice algorithm. This technique is based on the signal orthogonalization carried out by an adaptive lattice predictor in a previous stage. (ii) Gμ-FxSLMS: Filtered-x sequential least mean square algorithm with step-size gain. This strategy is based on partial updates of the weights of an adaptive filter as well as on the controlled increase in step size of the algorithm. This work illustrates by means of two different algorithms the trade-off established among computational costs, convergence rate, stability and mse excess when DSP-based strategies are used in control systems focused on the attenuation of acoustic disturbances.

Key words: Adaptive algorithms; active noise control; gradient adaptive lattice predictor; gain in step size; sequential partial updates.

1. ALGORITHMS

1.1 Modified Filtered-x Gradient Adaptive Lattice (FxGAL) Algorithm.

The FxGAL algorithm (Vicente et al., 2003) can be described as an extended version of the gradient adaptive lattice (GAL) algorithm (Griffiths, 1978) suitable to be used in the context of active control. The objective of

FxGAL and Modified FxGAL algorithms is to obtain faster and much less signal dependent convergence than that of FxLMS systems, while maintaining the numerical stability of stochastic gradient algorithms. Also, better tracking capabilities can be expected in non-stationary environments with the FxGAL algorithms. The price of these improvements is an increase in computational complexity, which can be easily lessened by reducing conveniently the order of the adaptive lattice orthogonalizer.

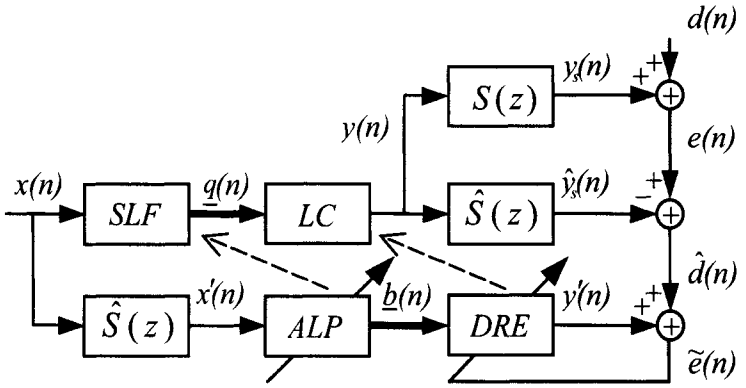


Figure 8-1. Block diagram of the modified FxGAL algorithm.

The modified version of the FxGAL algorithm makes use of the same idea that leads from the standard FxLMS to the Modified FxLMS[4] algorithm (Bjarnason, 1992; Kim et al., 1994): an estimation of the primary noise is used to properly swap the order between secondary path and adaptive control filter and a simultaneous copy of this control filter is used with the reference signal. In this way, the limitations imposed on the step size for the standard version of the algorithm are overcome in the modified one.

The key system in the FxGAL algorithms is an Adaptive Lattice Predictor (ALP), which realizes an approximate time-domain orthogonalization of its input data, without loss of information. The combination of this orthogonalization with independent absolute step sizes for each filter weight in the Desired Response Estimator (DRE) makes possible the expected increase in convergence speed.

The block diagram of the modified FxGAL algorithm is shown in Figure 8-1. As it can be seen, the filtered reference signal is the input to the ALP-DRE combination, while the reference signal goes through a Slave Lattice Filter (SLF) and a Linear Combiner (LC) to yield the control signal $y(n)$. The ALP and DRE blocks are the adaptive ones, while the coefficients of the

[4] This algorithm is also sometimes called Constraint FxLMS (Kim et al., 1994).

SLF and LC systems are simply copied from the ALP and DRE, respectively. An iteration of the Modified FxGAL algorithm is given by:

$t_0(n) = q_0(n) = x(n)$ /* Slave Lattice Filter (SLF) */

for $l = 1$ *to* $Lw - 1$ *do* /* Lw: Length of the filter */

$t_l(n) = t_{l-1}(n) + k_l(n) \cdot q_{l-1}(n-1)$ /* $\underline{t}(n)$: Forward output of SLF */

$q_l(n) = q_{l-1}(n-1) + k_l(n) \cdot t_{l-1}(n)$ /* $\underline{q}(n)$: Backward output of SLF */

end of for

$y(n) = \underline{w}^T(n) \cdot \underline{q}(n)$ /* $\underline{w}(n)$: Linear Combiner (LC) filter */

$\hat{y}_s(n) = \underline{\hat{s}}^T(n) \cdot \underline{y}(n)$ /* $\underline{\hat{s}}(n)$: Estimate of the Secondary path */

$\hat{d}(n) = e(n) - \hat{y}_s(n)$ /* $\hat{d}(n)$: Estimate of the primary noise */

$x'(n) = \underline{\hat{s}}^T(n) \cdot \underline{x}(n)$ /* Filtering of the reference */

$f_0(n) = b_0(n) = x'(n)$ /* Adaptive Lattice Predictor (ALP) * /

for $l = 1$ *to* $Lw - 1$ *do*

$f_l(n) = f_{l-1}(n) + k_l(n) \cdot b_{l-1}(n-1)$ /* $\underline{f}(n)$: Forward prediction errors * /

$b_l(n) = b_{l-1}(n-1) + k_l(n) \cdot f_{l-1}(n)$ /* $\underline{b}(n)$: Backward prediction errors */

/* Recursive power estimate * /

$$\hat{P}_l(n) = \beta_{ALP} \cdot \hat{P}_l(n-1) + (1 - \beta_{ALP}) \cdot \left(f_{l-1}^2(n) + b_{l-1}^2(n-1) \right)$$

/* Updating reflection coefficients $\underline{k}(n)$ */

$$k_l(n+1) = k_l(n) - \frac{\alpha_{ALP}}{\hat{P}_l(n)} \cdot \left(f_{l-1}(n) \cdot b_l(n) + b_{l-1}(n-1) \cdot f_l(n) \right)$$

end of for

$y'(n) = \underline{w}^T(n) \cdot \underline{b}(n)$ /* $\underline{w}(n)$: Desired Response Estimator (DRE) * /

$\tilde{e}(n) = \hat{d}(n) + y'(n)$

for $l = 0$ *to* $Lw - 1$ *do*

/* Recursive power estimate * /

$$\hat{P}_{bl}(n) = \beta_{DRE} \cdot \hat{P}_{bl}(n-1) + (1 - \beta_{DRE}) \cdot b_l^2(n)$$

/ * Updating DRE coefficients * /

$$w_l(n+1) = w_l(n) - \frac{\alpha_{DRE}}{\max\{\hat{P}_{bl}(n), P_{\min}\}} \cdot b_l(n) \cdot \tilde{e}(n)$$

end of for

where α_{ALP} and α_{DRE} are, respectively, the step sizes before normalizing of the Adaptive Lattice Predictor and the Desire Response

Estimator and β_{ALP} and β_{DRE} are the forgetting factors used to carry out the recursive estimate of the power.

1.2 Filtered-x Sequential Least Mean Square Algorithm with Step-Size Gain (Gμ-FxSLMS)

Partial updates algorithms (Douglas, 1997) update only a portion of the filter at each time instant in order to reduce their computational complexity. These algorithms suffer from one drawback: their convergence speeds are also reduced in the same proportion.

The Gμ-FxSLMS algorithm (Ramos et al., 2004) is aimed at reducing the computational costs of the control strategy without either incrementing the final misadjustment or slowing down the convergence speed. The block diagram of the Gμ-FxSLMS is shown in Figure 8-2.

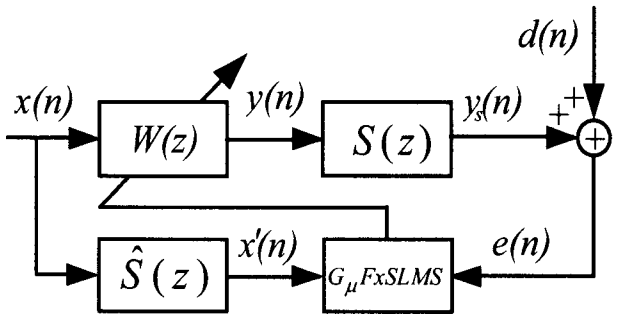


Figure 8-2. Block diagram of the Gμ-FxSLMS.

An iteration of the Gμ-FxSLMS algorithm can be expressed as follows:

$y(n) = \underline{w}^T(n) \cdot \underline{x}(n)$ /* *Generation of antinoise* */
if $n \bmod N == 0$ /* N : *Decimating factor* */
$x'(n) = \underline{\hat{s}}^T(n) \cdot \underline{x}(n)$ /* *Filtering with the estimate of the Secondary path* $\underline{\hat{s}}(n)$ */
end of if
for $l = 0$ *to* $Lw-1$ *do* /* Lw : *Length of the adaptive filter* */
if $(n\text{-}l) \bmod N == 0$ /* *Filter partial updates* */
/* μ : *Step size*; G_μ : *Gain in step size* */
$w_l(n+1) = w_l(n) - \mu \cdot G_\mu(N, Lw, Fs) \cdot e(n) \cdot x'(n-l)$
end of if
end of for

where the step-size gain Gμ is defined as the ratio between the bounds on the step sizes in two cases: firstly, when the adaptive algorithm uses sequential partial updates and, secondly, when every coefficient is updated at each iteration. In so doing, we obtain the factor by which the step size can be multiplied when the adaptive algorithm uses partial updates. This gain, that is approximately equal to the decimating factor N at most frequencies, allows the sequential strategy to achieve the convergence rate of the original FxLMS algorithm.

The theoretical analysis of the strategy prevents from the use of certain frequencies corresponding to notches which appear in the gain in the step size of the adaptive algorithm. Their width and exact location depend on the length of the adaptive filter (*Lw*), the decimating term *(N)* and the sampling frequency. Step-size gains for different values of the length of the adaptive filter and the decimating factor are shown in Figure 8-3. It can be easily derived from the examples given that the number of notches appearing in the gain is N-1. As far as the number of taps is concerned, the larger the adaptive filter is, the narrower the notch will be, that is, the narrower the bandwidth at which the gain in step size cannot be applied at its full strength will be.

To sum up, the step size can by multiplied by N in order to compensate the inherently slower convergence rate of the sequential adaptive algorithm as long as the regressor signal has no components at the notch frequencies.

2. COMPUTATIONAL COSTS

Table 8-1 shows the computational complexity of the Modified FxGAL and Gμ-FxSLMS algorithms when both strategies are used in the context of a two independent channel implementation of a feedforward ANC system.

Table 8-1. Computational complexity of the Modified FxGAL and the Gμ-FxSLMS algorithms in terms of the average number of additions, multiplies and divisions required per iteration.

Algorithm	# Additions	# Multiplications	# Divisions
Mod. FxGAL	$2\cdot(12\cdot Lw+2\cdot Ls-10)$	$2\cdot(17\cdot Lw+2\cdot Ls-10)$	$2\cdot(2\cdot Lw-1)$
Gμ-FxSLMS	$2\cdot\left(\left(1+\frac{1}{N}\right)\cdot Lw+\frac{Ls-1}{N}\right)$	$2\cdot\left(\left(1+\frac{1}{N}\right)\cdot Lw+1+\frac{Ls}{N}\right)$	*None*

where *Lw* is the length of the adaptive filter, *Ls* is the length of the off-line estimate of the secondary path and N is the decimating factor used in the partial updates of the second algorithm proposed.

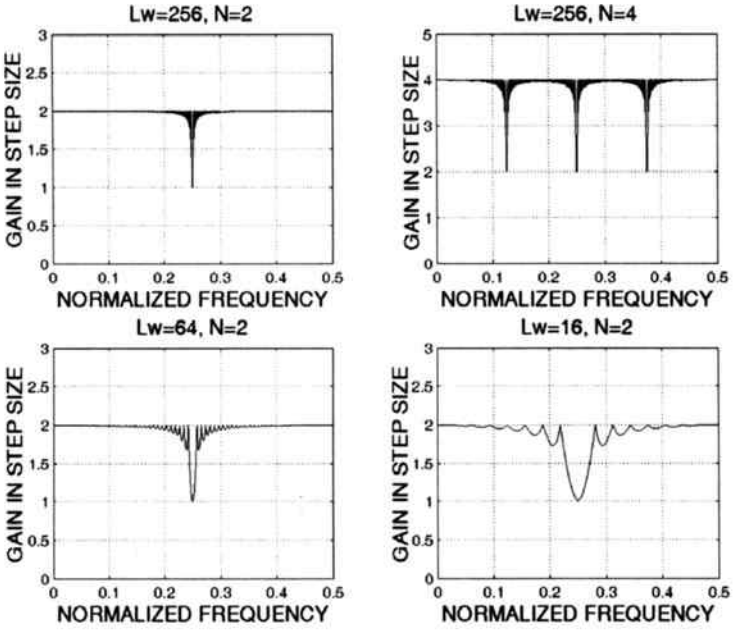


Figure 8-3. Step size gains for different values of adaptive filter length *Lw* and the decimation factor *N*.

It could be observed that the sampling frequencies chosen for the practical implementation of both strategies were not the same. While for the Modified FxGAL a sampling frequency of 1000 samples/s was considered to be enough to deal with low frequency noise, in the case of the Gμ-FxSLMS algorithm, the sampling frequency was set to a value 8 times higher, that is, 8000 samples/s in order to broaden the bandwidth free of notches in the step-size gain. As a result of that, the comparison between both strategies should be carried out on the basis of the number of operations required per second, instead of the number of operations per iteration.

3. EXPERIMENTAL RESULTS

3.1 Set-up

The physical arrangement of the electro-acoustic elements used in the implementation of the 1x2x2 Active Noise Control system placed at the front seats of a Nissan Vanette is depicted in Figure 8-4. The main Digital Signal Processor board employed to develop both strategies is the

PCI/C6600, based on the DSP TMS320C6701. The Input/Output board is the PMCQ20DS that disposes of 4 A/D and 4 D/A converters.

The control strategy implemented was either the Modified FxGAL algorithm or the Gμ-FxSLMS algorithm. In order to compare different control strategies, it is essential to repeat the experiment in the same conditions. So as to avoid fluctuations in level and frequency of the undesired disturbance, instead of starting the engine, we have previously recorded a signal consisting of two harmonics (150 and 450 Hz). The omnidirectional source Brüel & Kjaer Omnipower 4296 placed inside the van is fed with this signal and acts as the source of the primary noise.

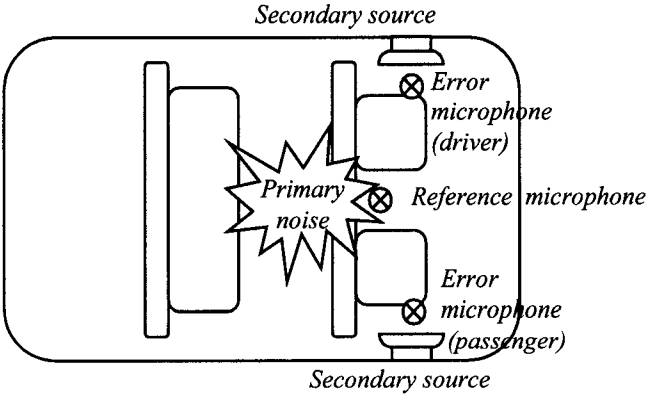


Figure 8-4. Physical disposal of the electro-acoustic elements inside the van.

3.2 Sets of Parameters Chosen

In order to obtain similar results with both algorithms in the attenuation of the undesired disturbance, the parameters are set to the following values:

- Modified FxGAL algorithm
  - Number of weights of the adaptive filter, $Lw = 8$.
  - Number of weights of the estimate of the secondary path, $Ls = 200$.
  - Sampling Frequency, $Fs = 1000$.
  - Normalized step size for the ALP stages is set to 0.06.
  - Forgetting factor, $\beta = 0.97$.
  - Normalized step size for the FxLMS algorithm is set to 0.08.

- Gμ-FxSLMS algorithm
  - Number of weights of the adaptive filter, $Lw = 128$.
  - Number of weights of the estimate of the secondary path, $Ls = 200$.
  - Sampling Frequency, $Fs = 8000$.
  - Decimating factor, $N = 8$.
  - Gain in step size, $G\mu = 8$;
  - Step size of the adaptive algorithm, $\mu = 0.1$.

So as to carry out a comparison of the computational requirements, it is assumed that the DSP can deal with 1 MAC operation -multiplication & accumulation- per DSP cycle whereas needs 40 cycles to perform a division (Poland, 1999).

According to the parameters chosen and taking into account the complexity expressed in Table 8-1, the number of clock cycles required between two consecutive samples is 2252 for the Modified FxGAL algorithm and 340 for the Gμ-FxSLMS algorithm. Considering that the sampling frequency is 8 times higher in the latter case, the cycles required per millisecond are 2252 and 2720, respectively. Thus, not only the performance achieved but also the computational costs of both strategies are quite similar despite being based on opposite underlying ideas.

3.3 Analysis in the Time Domain

Figures 8-5 and 8-6 show the learning curves of the Modified FxGAL and the Gμ-FxSLMS algorithms, respectively, when the error signals are measured at the microphones located near to the head of the driver and the passenger.

In both cases the two-harmonic signal is effectively attenuated by more than 20 dB within relatively short time.

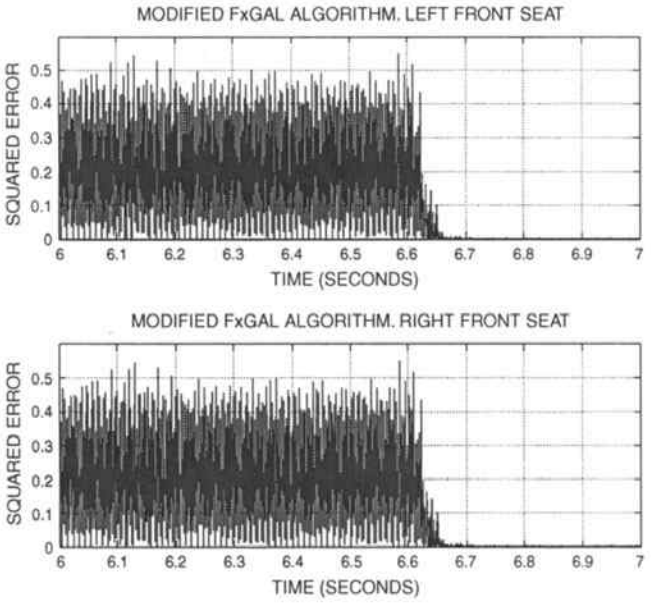


Figure 8-5. Evolution of the squared error when the ANC system based on the Modified FxGAL algorithm is switched on. Top: left front seat, bottom: right front seat.

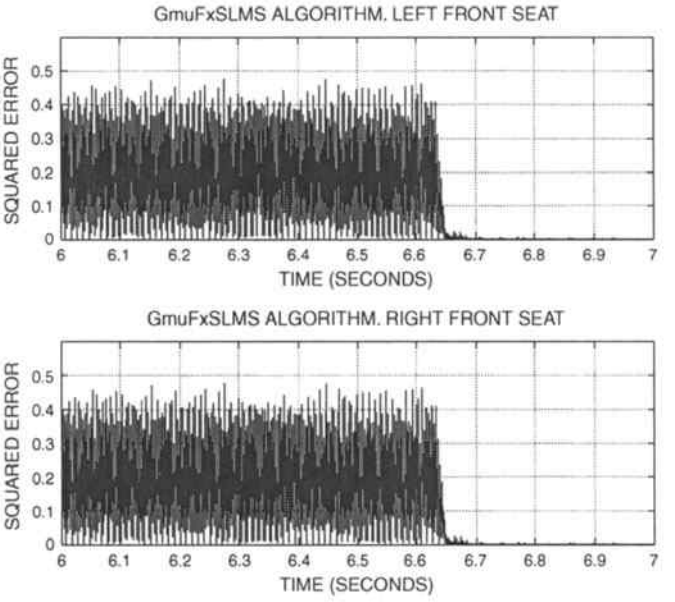


Figure 8-6. Evolution of the squared error when the ANC system based on the Gμ-FxSLMS algorithm is switched on. Top: left front seat, bottom: right front seat.

3.4 Analysis in the Frequency Domain

The frequency response functions measured at the error sensors located at the front seats of the van are shown in Figure 8-7 -Modified FxGAL algorithm- and Figure 8-8 -Gμ-FxSLMS algorithm-. The signal before control is shown in a dotted line whereas the signal after control is shown in a solid line. As far as the attenuation achieved is concerned, more than 25 dB of peak reduction are obtained at the main harmonics with both ANC algorithms. Nonetheless, very little off-peak reduction was obtained.

Power spectral density of the undesired noise depicted in Figures 8-7 and 8-8 consists of two harmonics at 150 and 450 Hz. Looking carefully into these graphs, it is to see that an unexpected noise component appears at a narrow frequency band between 15 and 35 Hz. Provided that this component was not present in the two-harmonic signal when it was generated, we can conclude that it corresponds to a mode imposed by the geometry of the van. In fact, we have verified that this low frequency noise vanishes as soon as the microphone is located outside the van.

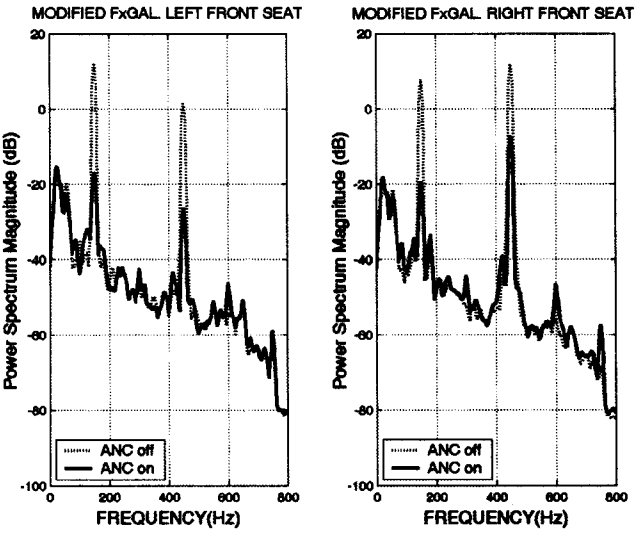


Figure 8-7. Experimental control results for the ANC system based on the Modified FxGAL algorithm in the frequency domain. Left: left front seat, right: right front seat.

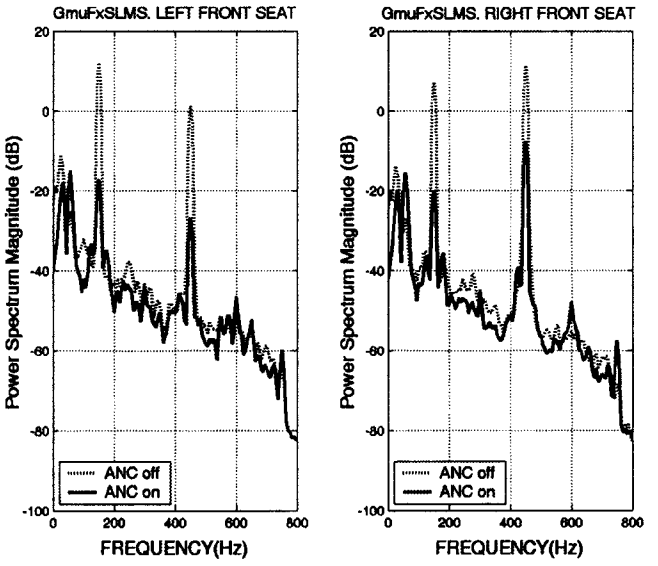


Figure 8-8. Experimental control results for the ANC system based on the Gμ-FxSLMS algorithm in the frequency domain. Left: left front seat, right: right front seat.

4. CONCLUSION

This chapter presents the results for applying two different control algorithms -Modified FxGAL and Gμ-FxSLMS- to actively attenuate periodic noise in a van.

The former strategy is aimed at speeding up the convergence rate at the expense of increasing the computational requirements whereas the latter puts forward a computationally less intensive solution without slowing down the convergence rate.

In spite of the fact that the underlying proposals of both algorithms are based on opposite control strategies -higher complexity and faster convergence rate versus lower complexity-, the subsequent choice of the parameters allows Modified FxGAL and Gμ-FxSLMS algorithms to achieve similar performance in terms of convergence speed, residual error and degree of attenuation with a computational complexity of the same order.

It has been experimentally shown that periodic noise may be substantially attenuated by Active Noise Control systems based on both algorithms.

With a sampling frequency of 1000 samples/s -for the Modified FxGAL algorithm- or 8000 samples/s -for the Gμ-FxSLMS-, the effective ANC

system bandwidth is approximately 500 Hz, and usually produces more than 20 dB of reduction within about 0.05 seconds of the starting.

ACKNOWLEDGEMENT

This work has been generously supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT) of Spain under grants TIC-2002-04103-C03-01 and TIN-2005-08660-C04-01.

REFERENCES

[1] Bjarnason, E., 1992, Active noise cancellation using a modified form of the filtered-x LMS algorithm, in: Proceedings of Eusipco-92, Brussels, pp. 1053–1056.

[2] Douglas, S.C., 1997, Adaptive filters employing partial updates, IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing **44** (3): 209–216.

[3] Griffiths, L. J., 1978, An adaptive lattice structure for noise-cancelling applications, in: Proceedings of ICASSP 1978, Tulsa, pp. 87–90.

[4] Kim, I.-S., Na, H.-S., Kim, K.-J., and Park, Y., 1994, Constraint filtered-x and filtered-u least-mean-square algorithms for the active control of noise in ducts, J. Acoust. Soc. Am. **95** (6): 3379–3389.

[5] Poland, S., 1999, TMS320C67xx Divide and Square Root Floating-Point Functions, Texas Instruments, App. Rep, SPRA516.

[6] Ramos, P., Torrubia, R., López, A., Salinas, A., and Masgrau, E., 2004, Computationally efficient implementation of an active noise control system based on partial updates, in: Proceedings of ACTIVE 2004, Williamsburg.

[7] Vicente, L., Masgrau, E., and Sebastián, J.M., 2003, Active noise control experimental results with FxGAL algorithm, in: Proceedings of Internoise 2003, Jeju Island, South Korea, October 2004.

Chapter 9

COMPARATIVE STUDIES ON SINGLE-CHANEL DE-NOISING SCHEMES FOR IN-CAR SPEECH ENHANCEMENT

Weifeng Li, Katunobu Itou, Kazuya Takeda, and Fumitada Itakura
Graduate School of Engineering, Graduate School of Information Science, Nagoya University; Faculty of Science and Technology, Meijo University Nagoya, 464–8603 Japan

Abstract: This chapter describes a novel single-channel in-car speech enhancement method that attempts to estimate the log spectra of speech with a close-talking microphone. It is based on the nonlinear regression of the log spectra of noisy signal captured by a distant microphone and the estimated noise. We compare the speech enhancement performance of proposed method to those based on spectral subtraction (SS) and short-time spectral attenuation (STSA). The method under consideration provides significant overall quality improvement in our subjective evaluation on the speech enhanced using the regression method. We have conducted isolated word recognition experiments over dataset from 15 real car driving conditions. The proposed adaptive nonlinear regression approach shows an improvement in average word error rate (WER), reductions of 54.2% and 16.5%, respectively, when compared to the original noisy speech and the ETSI advanced front-end experiments of [15].

Key words: Spectral subtraction; short-time spectral attenuation; multi-layer perceptron; mean opinion score; pair-wise preference test; speech recognition

1. INTRODUCTION

Among a variety of speech enhancement methods, techniques based on *spectral subtraction* (SS) [1] and *short-time spectral attenuation* (STSA) [2] [4] are commonly used. SS based methods generally make assumptions about the independence of speech and noise spectra, allowing for simple linear subtraction of the estimated noise spectra. Although scaling factors for

emphasis or de-emphasis of the estimated noise have been proposed to reduce "musical tone" artifacts in the SS method, the specifications of the scaling factors are usually obtained experimentally. On the other hand, STSA based methods tend lead to a nonlinear spectral estimator by introducing a priori SNR. However, they require assumptions about *ad-hoc* statistical distributions for speech and noise spectra [3] [4]. It is also well – known fact that both SS and STSA based methods can only handle additive noise.

In our previous work, we have proposed a novel and effective multi-microphone speech enhancement approach based on multiple regression of log spectra [5] that used a set of spatially distributed microphones. The idea was to approximate the log spectra of a close-talking microphone by effectively combining of the log spectra of distant microphones. The approach made no assumption about the positions of the speaker and noise sources with respect to the microphones and the computational load was very light. As it was reported in [6], this approach has been very effective based in our in-car speech recognition experiments.

In this chapter, we extend the idea to the single-microphone set-up and propose that the log spectra of the clean speech can be approximated through the nonlinear regression of the log spectra of the observed noisy speech and the estimated noise. The proposed approach, which can be viewed as a generalized log spectral subtraction, has the following properties:

1. It does not need any assumption concerning the statistical independence and distributions of the speech and the noise spectra.
2. It can handle a wide range of distortion models, rather than only additive noise.
3. Regression weights are obtained through statistical optimization. Once the optimal regression weights are trained in the learning phase, they are utilized to generate the estimated log spectra in the test phase, where clean speech is no longer required.

The objective of this chapter is to describe the proposed method and to demonstrate its effectiveness on speech enhancement and recognition in a vehicular environment. In Section 2, we present the proposed regression-based speech enhancement algorithm. In Section 3, we present our experiments and subjective test results on regression-enhanced speech. Next, we describe our speech recognition experiments using the proposed method in Section 4 and conclusions are drawn in Section 5.

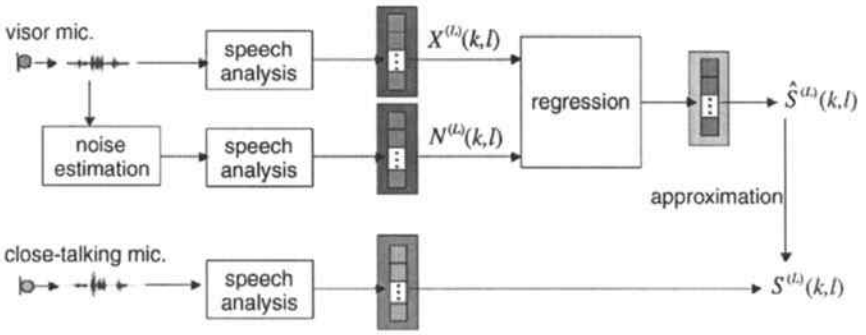


Figure 9-1. Schematic of the proposed regression-based speech enhancement.

2. REGRESSION-BASED SPEECH ENHANCEMENT

Let $s(i)$, $n(i)$, and $x(i)$ denote the reference clean speech (the speech at a close-talking microphone), the noise, and the observed noisy signal, respectively. By applying a window function and the analysis using short-time discrete Fourier transform (DFT), we have complex spectra $S(k,l)$, $N(k,l)$, and $X(k,l)$ of the clean speech the noise and the noisy speech, respectively. Here k and l denote the frequency bin and the frame indices. Taking the log of the amplitude, we obtain $S^{(L)}(k,l)$, $X^{(L)}(k,l)$, and $N^{(L)}(k,l)$:

$$\begin{aligned} S^{(L)}(k,l) &= \log|S(k,l)|, \\ N^{(L)}(k,l) &= \log|N(k,l)|, \\ X^{(L)}(k,l) &= \log|X(k,l)|. \end{aligned} \tag{1}$$

The idea of regression-based speech enhancement is to approximate $S^{(L)}(k,l)$ by combining $X^{(L)}(k,l)$ and $N^{(L)}(k,l)$ as shown in Figure 9-1. From the experiments in the multi-microphone experiments [6], we use a *multi-layer perceptron* (MLP) regression method for the regression system. The MLP has one hidden layer composed of eight (8) neurons. Therefore, the estimate of our clean speech $\hat{S}^{(L)}(k,l)$ can be calculated from $X^{(L)}(k,l)$ and $N^{(L)}(k,l)$ using

$$\hat{S}^{(L)}(k,l) = b_k + \sum_{p=1}^{8} w_{k,p} \tanh(f(X^{(L)}(k,l), N^{(L)}(k,l))), \quad (2)$$

where $\tanh(\circ)$ is the tangent hyperbolic activation function and

$$f(X^{(L)}(k,l), N^{(L)}(k,l)) = b_{k,p} + w_{k,p}^{x} X^{(L)}(k,l) + w_{k,p}^{n} N^{(L)}(k,l). \quad (3)$$

p is the index of the hidden neurons. The parameters, i.e., regression weights $\{b_k, w_{k,p}, w_{k,p}^{x}, w_{k,p}^{n}, b_{k,p}\}$ are obtained from minimizing the mean squared error (MSE):

$$\varepsilon(k) = \sum_{l=1}^{J} [S^{(L)}(k,l) - \hat{S}^{(L)}(k,l)], \quad (4)$$

through the back-propagation algorithm [7]. Here, J denotes the number of training examples (frames).

Once $\hat{S}^{(L)}(k,l)$ is obtained for each frequency bin, enhanced speech can be generated by taking the exponent and performing a standard short-time inverse discrete Fourier transform (IDFT) with the combination of the phase of the observed noisy speech.

The proposed approach is cast into a single-channel methodology because once the optimal regression parameters are obtained by regression learning, they can be utilized to generate $\hat{S}^{(L)}(k,l)$ in the test phase, where the speech of the close-talking microphone is no longer required. Multiple regression in this context means that regression is performed for each frequency bin. The use of minimum mean squared error in the log spectral domain is motivated by the fact that the log spectral measure is more related to the subjective quality of speech [8] and that numerous impressive results have been reported in speech recognition applications based on log distortion measures [9] [10]. Although, both the proposed regression-based method and *log-spectra amplitude* (LSA) estimator [4] employ minimum mean squared errors (MMSE) as their cost function in the log domain, the former does not make any assumption regarding the distributions of speech and noise spectra. The proposed method differs from [10] in that it does not need to estimate the mean and the variance of log spectra of the clean speech, which is non-trivial because only noisy speech is available. Moreover, the proposed method employs more general regression models and has less latency.

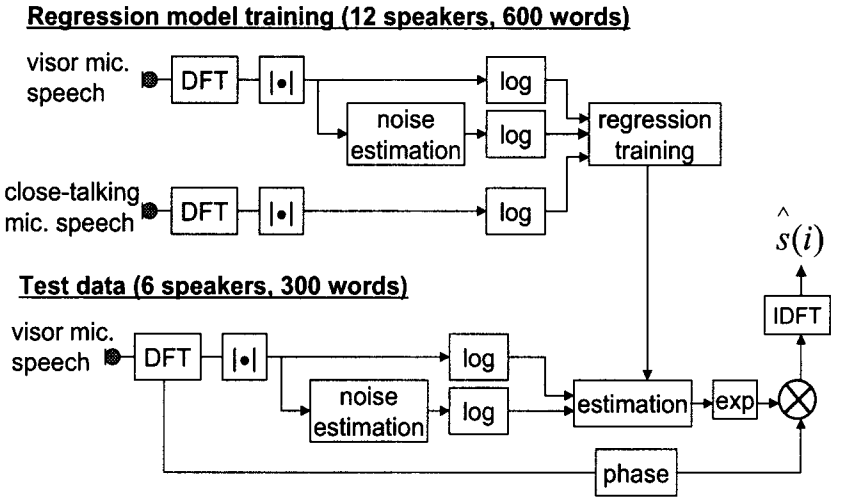


Figure 9-2. Diagram of regression-based speech enhancement.

3. EXPERIMENTS AND SUBJECTIVE TESTS

3.1 Experimental Setup

The speech data used for evaluating the performance of the proposed technique are from the CIAIR in-car speech corpus [16]. We have used a microphone at the visor position and a close-talking microphone (with a headset) in our experiments. The close-talking microphone is used as the reference speech for training the regression system. The test speech was based on 50 isolated-word sets under seven (7) real driving conditions listed in Table 9-1. Figure 9-2 shows a block diagram of the regression-based speech enhancement system for a particular driving condition. For each driving condition, the 50 isolated-word data uttered by each and every one of 12 speakers were used for learning the regression weights, and the remaining 300 words from another group of six speakers (three male and three female) were used for open testing.

Table 9-1. Seven driving conditions for speech enhancement evaluation.

Driving environment	in-car state
city driving	normal
city driving	CD player on
city driving	air-conditioner on at high level
city driving	window open
idling	normal
expressway driving	normal
expressway driving	window open

For comparison, a *parametric formulation of the generalized spectral subtraction* (PF-GSS) [11] and a *log-spectra amplitude* (LSA) estimator [4] were also investigated. For PF-GSS a constrained version, which was suggested by the authors, was used. An *a priori* SNR was calculated by the well-known "decision-directed" approach. An *improved minima controlled recursive averaging* (IMCRA) method [12] was employed for estimating the noise for all three methods. We selected PF-GSS and LSA because they can provide good noise reduction and reduce the annoying "musical tone" artifacts of enhancement schemes based on conventional spectral subtraction while maintaining relatively low computational complexity. Four different types of speech included in the evaluation are listed in Table 9-2.

Table 9-2. Four types of speech used for the evaluation.

Original	original noisy speech with no processing
PF-GSS	Speech enhanced using PF-GSS method
LSA	Speech enhanced using LSA method
Regression	Speech enhance using the proposed method

3.2 Procedure

For each driving condition, five speech samples were randomly selected from the ensemble of 300 test signals and the above mentioned four de-noising algorithms are applied. So formed 140 word utterances are presented to each listener.

Twelve test listeners (eight male and four female students aging from 19 to 28 years) participated in the evaluations. They had no prior experience in psycho-acoustic measurements and no history of hearing problems. They were seated in a sound-proof booth. Signal presentation was controlled by a computer. Signals were fed to listeners via a Sony-dynamic stereo headphone (MDR-CD900ST). Presentation level was individually adjusted

so that perception was "loud but still comfortable" to guarantee that most signal parts were audible to the listener.

In speech community, *Mean Opinion Score* (MOS) is frequently used as a reliable and easily implemented subjective performance measure. In this method, human listeners rate test speech on a five-grade scale. To minimize the inherent subjective judgment bias in MOS, Hansen and Pellom suggested incorporating a subjective *Pair-wise Preference Test* (PPT) [13]. In PPT, a series of pair-wise randomized processed signals are presented, and listeners simply select the one they prefer. An advantage of PPT over MOS is its ease for subjects and the elimination of judgment bias [14].

We have performed both the MOS test and the PPT on measuring the overall quality. In the MOS, listeners rated the speech signals on a five-grade test based on Absolute Category Rating (ACR). The four kinds of speech signals, which were randomly arranged, were presented as one measurement block. To adjust the rating differences, listeners evaluated speech signals corrupted by different noise levels and processing artifacts at the beginning of the subjective quality assessment. In the PPT evaluations, six pair-wise comparisons were presented as one measure block. Listeners were asked to state a preference for one of the two presented algorithms.

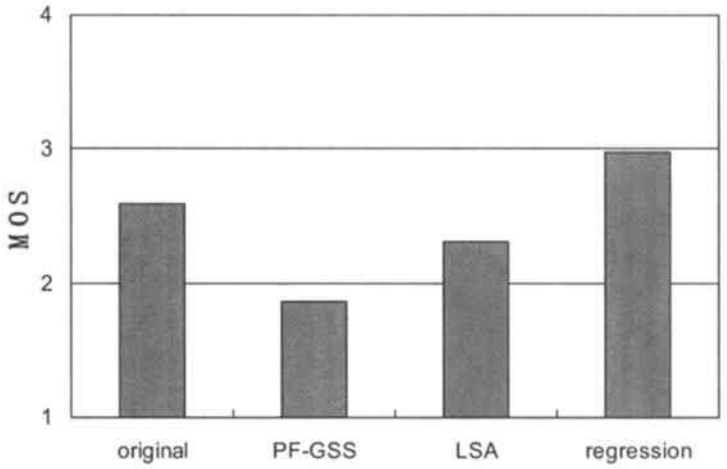


Figure 9-3. MOS (averaged over seven driving conditions).

3.3 Results

MOS results for these four algorithms averaged over seven driving conditions are displayed in Figure 9.3. It is found that the MOS results of PF-GSS and LSA are lower than the original noisy speech. This clearly indicates that PF-GSS and LSA enhancement methods seem to decrease overall speech quality rather than to increase. This is in line with the results of most publications (e.g., [14]) on single-microphone speech enhancement schemes. Compared to PF-GSS, the LSA algorithm has yielded higher MOS for the less "musical tone" artifacts introduced, while the regression-based enhancement method resulted in somewhat higher MOS values.

Table 9-3. Preference rates between algorithms.

	original	PF-GSS	LSA	regression
Original	0	75.48%	51.67%	31.43%
PF-GSS	24.52%	0	23.10%	10.24%
LSA	48.33%	76.90%	0	25.00%
Regression	68.57%	89.76%	75.00%	0

The PPT results are shown in Table 9-3. Scores in each row, which were calculated as vote percentages, denote the preference rates of one algorithm to another one. As expected, neither the PF-GSS nor LSA methods are preferred over the original observed speech. LSA gives higher preference scores compared to PF-GSS. Finally, regression-based enhancement method achieves significantly higher preference than all other algorithms, which clearly demonstrates the superiority of the proposed method.

4. SPEECH RECOGNITION EXPERIMENTS

We performed in-car speech recognition experiments using regression methods. Test size of the data has been extended to 50 words under all of the 15 real car driving conditions, as listed in Table 9-4. 1,000-state tri-phone Hidden Markov Models (HMM) with 32 Gaussian mixtures per state were used for acoustical modeling. They were trained over a total of 7,000 phonetically-balanced sentences collected from the visor microphone (3,600 during the idling-normal condition, and 3,400 while driving on the streets near Nagoya university (city-normal condition)). The feature set is a

25-dimensional vector consisting of 12 CMN-MFCC and 12 ΔCMN-MFCC and Δlog energy.

The above regression algorithms are implemented for each log mel-filter bank (MFB) output. The block diagram of in-car regression-based speech recognition for a particular driving condition is given in Figure 9-4. Once the estimated log MFB output is obtained for each mel-filter bank, the estimated log MFB vectors are transformed into mean normalized mel-frequency cepstral coefficients (CMN-MFCC) for recognition.

Table 9-4. 15 driving conditions (3 driving environments 5 in-car states).

driving environment	in-car state
idling city driving expressway driving	Normal CD player on air-conditioner on at low level air-conditioner on at high level window open

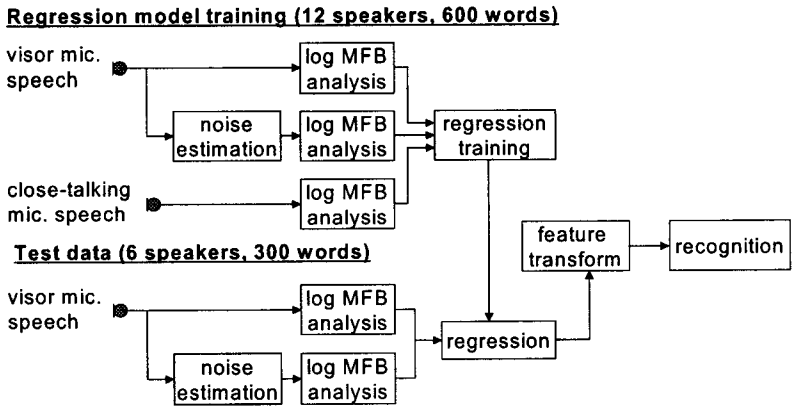


Figure 9-4. Diagram of regression-based speech recognition.

For comparison, we have also performed recognition experiments using a linear regression method and ETSI advanced front-end experiments [15]. In the linear regression method, no hidden layer (neurons) was used. The acoustical model used for ETSI advanced front-end experiments was trained over the training data processed with ETSI advanced front-end. The recognition performance averaged over the 15 driving conditions is given in Figure 9-5. It is found that all the enhancement methods outperform the original noisy speech. LSA gives higher recognition accuracy than PF-GSS.

ETSI advanced front-end very marginally outperforms LSA. Although linear regression is less effective than the conventional enhancement methods, nonlinear regression achieves the best recognition performance, outperforming ETSI advanced front-end by about 1.8 percent.

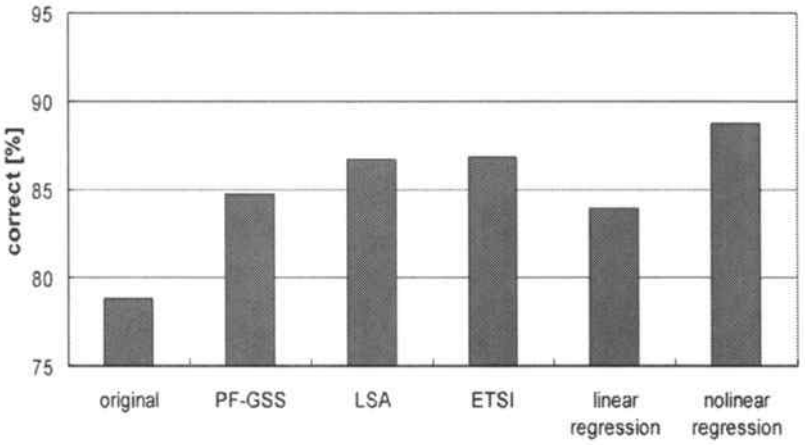


Figure 9-5. Diagram of regression-based speech recognition.

5. CONCLUSION

A novel regression-based speech enhancement method was proposed, which approximates the log spectra of clean speech with the inputs of the log spectra of noisy speech and estimated noise. The proposed method employs statistical optimization and makes no assumption about the independence nor the distributions of the speech and noise spectra. The proposed method provided consistent improvements in our subjective evaluation of regression-enhanced speech. The results of our studies on isolated word recognition under 15 real car driving conditions show that the proposed method outperforms conventional single-channel speech enhancement algorithms. Other methods for speech enhancement may be combined with the proposed method to obtain improved recognition accuracy in noisy environments. This method is expected to enhance recognition accuracy in very noisy situations and to be applicable to a large number of real-life applications.

ACKNOWLEDGEMENT

This work is partially supported by a Grant-in-Aid for Scientific Research (A) (15200014).

REFERENCES

[1] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-27, no.2, pp.113-120, 1979.

[2] O. Cappe and J. Laroche, "Evaluation of short-time spectral attenuation techniques for the restoration of music recordings," IEEE Trans. Speech and Audio Processing, vol.3, no.1, 1995.

[3] R. Martin, "Speech Enhancement Using MMSE Short Time Spectral Estimation with Gamma Distributed Speech Priors," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp.253-256, 2002.

[4] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-33, no.2, pp.443-445, 1985.

[5] W. Li, T. Shinde, H. Fujimura, C. Miyajima, T. Nishino, K. Itou, K. Takeda, and F. Itakura, "Multiple regression of log spectra for in-car speech recognition using multiple distributed microphones," IEICE Trans. on Information & Systems, E88-D, no.3, pp.384-390, 2005.

[6] W. Li, K. Itou, K. Takeda, and F. Itakura, "Optimizing regression for in-car speech recognition using multiple distributed microphones," Proc. International Conference on Spoken Language Processing, pp.2689-2692, 2004.

[7] S. Haykin, Neural Networks - A Comprehensive Foundation, Prentice-Hall, 1999.

[8] S.R. Quackenbush, T.P. Barnwell, and M.A. Clements, Objective Measures of Speech Quality, Prentice-Hall, 1988.

[9] J. E. Porter and S. F. Boll, "Optimal estimators for spectral restoration of noisy speech," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.18.A.2.1-18.A.2.4, 1984.

[10] F. Xie and D.V. Compernolle, "Speech enhancement by spectral magnitude estimation - A unifying approach," Speech Communication, vol.19, pp.89-104, 1996.

[11] B.L. Sim, Y.C. Tong, J.S. Chang, and C.T. Tan, "A parametric formulation of the generalized spectral subtraction method," IEEE Trans. Speech and Audio Processing, vol.6, no.4, pp.328-337, 1998.

[12] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging," IEEE Trans. Speech and Audio Processing, vol.11, no.5, pp.466-475, 2003.

[13] J.H.L. Hansen and B.L. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," Proc. International Conference on Spoken Language Processing, pp.2819-2822, 1998.

[14] M. Marzinzik, Noise reduction schemes for digital hearing aids and their use for the hearing impaired, Ph.D. thesis, University of Oldenburg, 2000.

[15] "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithm," ETSI ES 202050 v1.1.1, 2002.

[16] N. Kawaguchi, S. Matsubara, I. Kishida, Y. Irie, H. Murao, Y. Yamaguchi, K. Takeda and F. Itakura, "Construction and Analysis of the Multi-layered In-car Spoken Dialogue Corpus," Chapter 1 in *DSP in Vehicular and Mobile Systems,* H. Abut, J. H.L. Hansen, and K. Takeda (Editors), Springer, New York, NY, 2005.

Chapter 10

ADVANCES IN ACOUSTIC NOISE TRACKING FOR ROBUST IN-VEHICLE SPEECH SYSTEMS[5]

Murat Akbacak and John H.L. Hansen
Center for Robust Speech Systems, University of Texas at Dallas, Richardson, Texas, USA
Email: {Murat.Akbacak,John.Hansen}@utdallas.edu

Abstract: Speech systems work reasonably well under homogeneous acoustic environmental conditions but become fragile in practical applications involving real-world environments (e.g., in-car, broadcast news, digital archives, etc.) where the audio stream contains multi-environment characteristics. To date, most approaches dealing with environmental noise in speech systems are based on assumptions concerning the noise, rather than exploring and characterizing the nature of the noise. In this chapter, we present our recent advances in the formulation and development of an in-vehicle environmental sniffing framework previously presented in [1,2,3,4]. The system is comprised of different components to detect, classify and track acoustic environmental conditions. The first goal of the framework is to seek out detailed information about the environmental characteristics instead of just detecting environmental change points. The second goal is to organize this knowledge in an effective manner to allow intelligent decisions to direct subsequent speech processing systems. After presenting our proposed in-vehicle environmental sniffing framework, we consider future directions and present discussion on supervised versus unsupervised noise clustering, and closed-set versus open-set noise classification.

Key words: Automatic speech recognition, robustness, environmental sniffing, multi-modal, speech enhancement, model adaptation, environmental sniffing, dialog management, mobile, route navigation, in-vehicle.

[5] 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.

1. INTRODUCTION

Significant advances in speech technology have been achieved in applications where the environmental condition is homogeneous. Most recently, research has shifted to real-world environments where changing environmental conditions represent significant challenges in maintaining speech system performance. 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). These applications present research challenges due to the wide diversity of acoustic environmental conditions and the need to maintain near-real time performance.

The problem of changing environmental conditions has seen considerable attention especially in Automatic Speech Recognition (ASR) applications since recognition performance degrades substantially due to changes in the environment [5, 6, 7, 8]. All efforts in the field of noisy speech recognition have been directed at reducing the performance mismatch between training and operating conditions. These techniques are grouped into the following categories:

1. Re-Training & Multi-Style Training
2. Speech Enhancement & Feature Enhancement
3. Noise Resistant Features
4. Model Adaptation

In the re-training method, an "environment-dependent" system is re-trained with data from new testing environments. The main disadvantage of this technique is the lack of *a priori* knowledge of environmental characteristics. In addition to this, data collection and transcription is time consuming and the training process is extremely computationally expensive.

In multi-style training, an "environment-independent" system is trained by pooling data from different acoustical environments. The disadvantage of this method is the lack of sufficient environments needed to achieve environment independence. Also, it is unclear how speech from diverse environmental conditions contributes to the overall speech recognition model.

Most early work towards robustness has been derived from classical techniques developed in the context of speech enhancement ([9] offers a good historical summary, and [10] represents a more recent summary on enhancement techniques). The goal is to transform noisy speech into a

reference environment, and recognize it with a system trained in the reference environment. Speech enhancement methods offer the distinct advantage of requiring no training data, and can be enabled or disabled with limited changes to the subsequent speech task.

In the robust features method, it is assumed that the system is noise independent, and uses the same system configuration for both noisy and clean speech recognition. The goal is to derive noise resistant parameters. One of the advantages of this technique is that in general weak or no assumptions are made about the noise (i.e., no explicit estimation of the noise statistics is required). On the other hand, this could be a shortcoming since it is impossible to make full use of characteristics specific to a particular noise type.

Model adaptation schemes transform the speech models created in the reference environment in order to accommodate the evolving noisy environment. Since accurate estimates of the noise statistics are required, this method can be sensitive to varying SNR (signal-to-noise ratio) and non-stationary noise environments.

While speech or feature enhancement, robust features, or model adaptation can be effective for robust speech recognition in noise, it is difficult to outperform a system that is trained in the same noise type and level when noisy conditions are stationary. While speech enhancement and model adaptation methods typically have access to a short segment of noise for statistical characterization, a full re-training approach typically requires several hours of speech in noise data. As such, many ASR researchers have migrated towards dedicated trained systems.

In more recent studies, as computational power has increased with the help of high-speed computers, a parallel bank of recognizers has been used in a Recognizer Output Voting Error Reduction (ROVER) paradigm [11]. This method seeks to reduce word error rates for dedicated ASR engines by exploiting differences in the nature of the errors made by multiple speech recognizers which use different features in the feature extraction step, different noise compensation schemes in the enhancement step, or different model adaptation schemes. The disadvantage of this method is the high computational power it requires, making it less feasible for real-time or dialog applications. In addition, it is not a general solution for other speech systems (e.g., speech coding, speech enhancement, etc.).

In this chapter, we present an overview of Environmental Sniffing framework previously presented in [1,2,3,4] with current extensions to the system. This chapter is organized as follows. In Section 2, we present our proposed framework. In Section 3, we present the algorithm formulation and evaluation results. In Section 4, we consider future directions. Sections 5 and 6 conclude with a discussion and a summary, respectively.

2. GENERAL SYSTEM ARCHITECTURE

A proposed general system architecture diagram for Environmental Sniffing is shown in Figure 10-1. Digitized speech is denoted as *s(n)*, captured from an input sensor (i.e., single or multi-microphone) and acoustic environmental information as *I(n)* which is a function of the input signal.

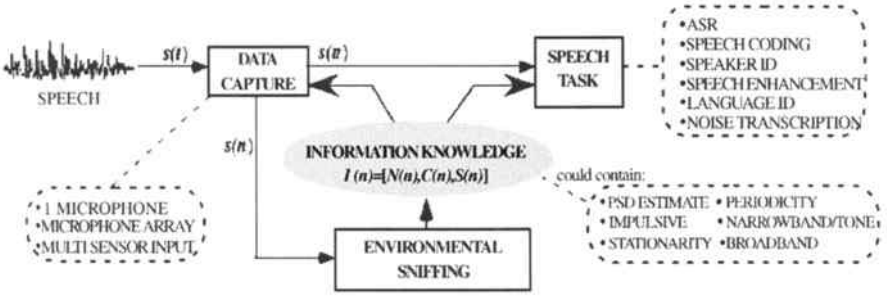


Figure 10-1. Operational schematic of the proposed Environmental Sniffer.

In a sample scenario, *s(n)* may be the audio data obtained in a vehicle with a microphone array, the speech task may include model adaptation within an ASR system, and *I(n)* may consist of the existing noise types with time tags and the power spectral estimates of the environmental noise with a stationarity measure. Here, *I(n)* could also contain a suggestion to use one of several adaptation schemes (Jacobian adaptation, MLLR [16], PMC [17], etc.), or alternative parameterization (MFCC, LPC, PLP [18], SBC [19], WPP [20], etc. which gives the best performance for the environmental noise knowledge estimated through Environmental Sniffing.

In addition to environmental noise knowledge *N(n)*, *I(n)* may contain *S(n)* - knowledge about the speaker identity and *C(n)* - channel information for the speech tasks having multi-*SEC* (*S*peaker, acoustic *E*nvironment, *C*hannel) characteristics. Knowledge from *S(n)* can be used to monitor the speaker's speaking style. It may consist of the accent and stress levels of the speaker so that correct pronunciation and duration modeling or acoustic model compensation techniques can be employed. This knowledge may be used in speech coding to improve the naturalness of speech. *C(n)* may provide the knowledge of channel type (bandwidth, type of distortion, etc.), impact of channel (fade-out, burst-error, channel bias, etc.) to improve the ASR system performance by having reliable parameters for feature enhancement or model adaptation.

So far we focused on cases where knowledge *I(n)* extracted from the speech signal is used to improve performance of speech systems dealing directly with the acoustic signal *s(n)*. *I(n)* can also be used to extract

information to have better user modeling within the context of dialog management. As an example, if *S(n)* includes high-level Lombard effect or emotion, the dialogue system might become more user-initiative, or if *I(n)* detects that the music is on in the car, the dialogue system can ask the user to turn it off, or warn the user that the navigation system would not function well. Integrating multiple sources of information (e.g., acoustic, lexical, nonverbal) within the dialogue management would make the navigation system more user friendly.

In the remainder of this study, we focus on changing acoustic environmental conditions and construct the *Environmental Sniffing* framework to extract environmental noise knowledge *N(n)* from an input audio stream. In other words, *I(n)* will consist of only environmental noise knowledge *N(n)*, with a constant channel- *C(n)* and speaker- *S(n)* traits.

3. ENVIRONMENTAL NOISE SNIFFING

In [1], we focused on extracting knowledge concerning the acoustic environmental noise using a noise-only audio database containing 8 noise conditions in a car environment. In [2], we presented a broad class monophone recognition based system for sniffing noisy-speech data, as shown in Figure 10-2, and we turned to a specific solution for an in-vehicle digit recognition system. In addition to 8 noise conditions [N1-N7, NX], the acoustic condition set contains also the clean condition- CL as presented in [1,2,3,4].

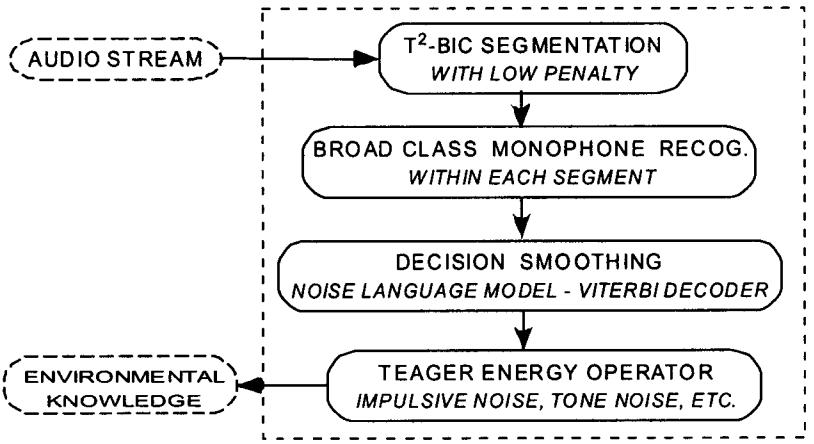


Figure 10-2. Environmental Noise Sniffing within noisy speech.

As Figure 10-2 shows, the incoming audio stream is first segmented into acoustically homogeneous speech blocks using our T^2-BIC [12,13] segmentation scheme with a low false alarm penalty (i.e. false alarms are tolerable to ensure we capture all potential marks, both true and false). For each segment, a lattice is generated in an FST (Finite State Transducer) format via phoneme recognition. During decision smoothing, the resulting phone-lattice of each segment is combined with an FST representing the noise language model to recover segmentation and classification errors.

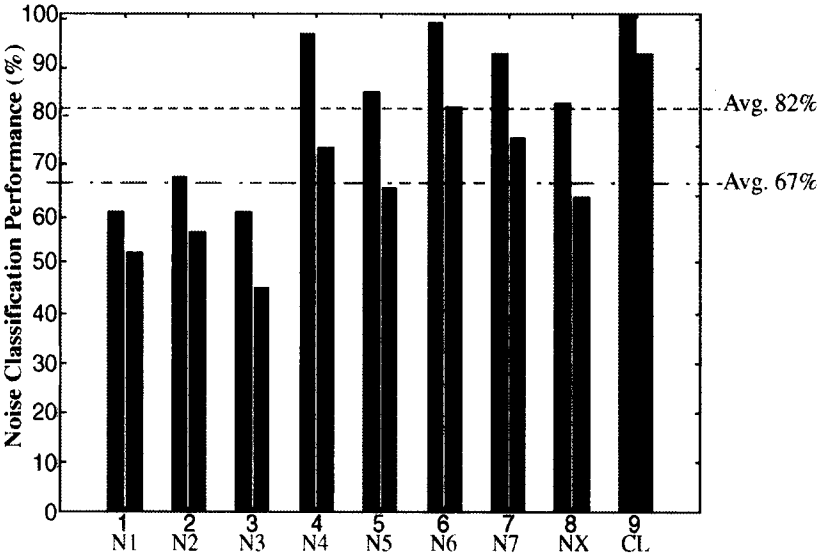


Figure 10-3. Environmental Noise Sniffing performance.

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 [1]. 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 10-2, 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 as shown in Figure 10-3. Environmental condition-specific acoustic models were trained and used during recognition tests.

Having established the environmental sniffer for directing ASR model selection, we now turn to ASR system evaluation. We tested and compared the following three 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.

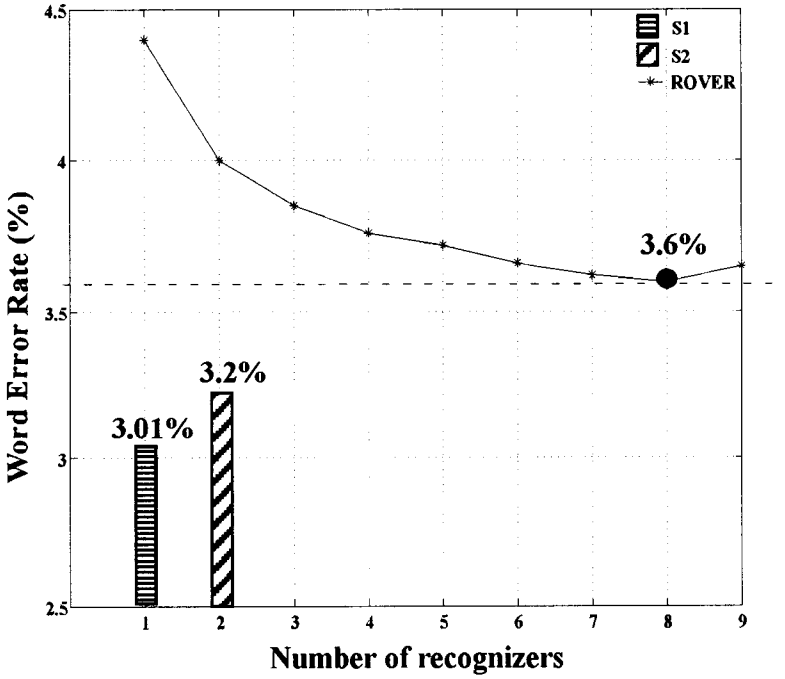


Figure 10-4. Word Error Rates for Digit Recognition Tests: S1 – matched noise model case, S2 – environmental sniffing model selection (1 CPU for sniffing, 1 CPU for ASR), S3 (ROVER) – employs up to 9 recognizers (i.e., CPUs) trained for each noise condition with ROVER selection.

As Figure 10-4 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 10-4, 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.

4. FUTURE DIRECTIONS

4.1 Supervised vs. Unsupervised Noise Clustering

Here, a supervised training process with pre-defined noise types was employed for in-vehicle noise clustering and classification. We did this in order to tag noise events for in-vehicle speech dialog systems. 15 noise classes are transcribed during the data collection by a transcriber seated in the car. A procedure was used where the time tags were generated instantly by the transcriber. After data collection, several noise conditions were grouped together, resulting in 8 acoustically distinguishable noise classes.

An alternative approach would be to consider an unsupervised noise clustering method with no prior noise type list. To do this, here, we use BIC (Bayesian Information Criteria) [12] based segmentation and clustering to obtain noise entries without associated physical events. In applications (e.g., digital archives, etc.) where it is not feasible to manually segment and label the different noise types (i.e., large number of noise types might result in inconsistent human labeling), unsupervised acoustic noise analysis gains importance.

After extracting homogeneous noise segments via BIC based segmentation, we use agglomerative bottom-up BIC based clustering to merge acoustically similar noise segments. Initially, each segment is a cluster by itself and the clusters are modeled by a single Gaussian. At each step of the clustering algorithm, a similarity measure is calculated for each pair of clusters. The two closest clusters are merged if the corresponding BIC variation is negative. If the difference is positive, the algorithm is stopped.

Environmental Sniffing with unsupervised noise analysis module would be also useful for automatic transcription of noisy speech corpora where the accuracy is much lower than that of transcription of clean speech. If we consider the fact that no standards exist for noise transcription in audio material, it is understandable that no study has yet considered an intelligent sniffing solution. The broad grouping of Broadcast News material [5, 6, 7, 8] represents the only large scale corpus that considers different acoustic

conditions, yet it is not directly labeled based on specific noise events. As the need for reliable speech systems in adverse conditions continues to grow, it becomes critical to automatically transcribe environmental noise with high accuracy for more effective speech system training.

4.2 Closed-set vs Open-set Noise Classification

Noise subspaces such as cell phone environment S^{cell}, in-vehicle environment S^{car}, or spoken document retrieval environment S^{sdr} can be constructed as shown in Figure 10-4. Within the in-vehicle environment S^{car}, among the 8 noise classes used in [1,2,3,4], some of the noise classes (e.g., windows open in-city traffic) occupy bigger acoustic space than the others (e.g., windows closed), and have bigger potential to introduce new noise types. Since we used a limited number of noise classes, we introduced noise class NX to represent open-set noise class. In other words, in addition to the closed member noise types that exist in the set, S^{car} will also possess a complementary noise class which represents the open out-of-set (OoS) noise types.

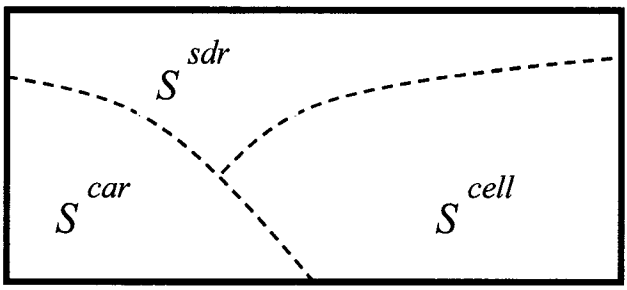


Figure 10-5. Noise subspaces for different tasks.

To construct the task dependent noise space, we start with a known initial set of noise types using training data collected for the specific task in either supervised or unsupervised way. After using this initial set on the development test data, we update the set by either adding new noise types or removing noise types if they do not occur.

Using the unsupervised noise clustering module presented in Section 4.1, we cluster the data classified as out-of-set (OoS) to see if there are any additional new noise type sets. Whenever there is sufficient training data for a noise cluster within NX, we update our noise list, and train an acoustic model to represent the new noise type.

For open-set noise classification (i.e., whether an incoming noise segment belongs to in-set or out-set NX), we use Universal Background

Model (UBM) based approach which is previously used for in-set/out-of-set speaker recognition in [21].

5. DISCUSSION

For our experiments in Section 3, we considered evaluation results for noise classification where Environmental Sniffing is employed to find the *highest* probable noise type. This is most appropriate for the goal of determining a single solution for the speech task problem at hand (e.g., selecting one ASR engine, a single feature set, etc.). However, in some speech applications, it may be useful to find the n least probable acoustic noise types among N acoustic noise conditions (i.e., "can we rule out n/N of the possible noise conditions?"). This could be useful for reducing speech recognizer computation in a ROVER paradigm if certain noise trained ASR engines could be eliminated, or in cases where we have an open noise condition where the noise at hand is not part of the set of N noise types and removing very low probability noise types could help in subsequent speech processing tasks. Finally, in an ASR task, even a ROVER paradigm could take advantage of Environmental Sniffing to address open questions such as the combination order of the ASR system hypotheses (i.e., weighting recognizer outputs based on sniffing scores), selecting the number of systems that should be combined, to engage or disable preprocessing or normalization of system outputs prior to combination, etc. Table 10-1 shows noise classification performance for different noise sets. For each noise set, we can employ ROVER paradigm with changing numbers of recognizers.

These alternative points of view all suggest that Environmental Sniffing can be helpful in system operation. The goal therefore has been to emphasize that direct estimation of environmental conditions should provide important information to tailor a more effective solution to robust speech processing systems.

Table 10-1. Merging confusable noise classes.

Noise classes	*Error*	*Accuracy*
{N1,N2,N3,N4,N5,N6,N7,NX}	0.00%	100.00%
{N1,N2,N3,N4,N5,N6,N7},{NX}	1.80%	98.12%
{N1,N2,N3},{N4,N5,N6,N7},{NX}	4.40%	95.60%
{N1,N2,N3},{N4},{N5,N6,N7},{NX}	7.39%	92.61%
{N1,N2,N3},{N4},{N5,N7},{N6},{NX}	10.75%	89.25%
{N1,N2,N3},{N4},{N5},{N6},{N7},{NX}	12.62%	87.38%
{N1},{N2,N3},{N4},{N5},{N6},{N7},{NX}	15.75%	84.25%
{N1},{N2},{N3},{N4},{N5},{N6},{N7},{NX}	25.51%	74.49%

6. CONCLUSION

In this chapter, we have addressed the problem of characterizing changing acoustic environmental conditions for improving speech system performance. We previously proposed framework, *Environmental Sniffing,* to detect, classify and track changing acoustic environmental conditions. The extracted knowledge can include noise, channel, and speaker traits. After proposing a general framework, we specialized the sniffer to an in-vehicle speech application. Novel aspects include a number of knowledge based processing steps such as T^2-BIC segmentation, noise language modeling, and GMM/HMM based classification. Experiments showed that the sniffing framework provides important information regarding changing acoustic conditions necessary for ASR applications that require near real-time performance such as in-vehicle dialog systems. We believe such processing could provide significant knowledge to help direct and improve subsequent speech processing tasks and thereby increase robust speech systems performance.

REFERENCES

[1] M. Akbacak, J. H. L. Hansen, "Environmental Sniffing: Noise Knowledge Estimation for Robust Speech Systems," IEEE ICASSP-2003: International Conference Acoustics Speech & Signal Processing, vol. 2, pp. 113-116, Hong Kong, April 2003..

[2] M. Akbacak, J. H. L. Hansen, "Environmental Sniffing: Robust Digit Recognition for an In-Vehicle Environment", Interspeech-Eurospeech-2003, pp.2177-2180, Geneva, Switzerland, September 2003.

[3] M. Akbacak, J. H. L. Hansen, "Environmental Sniffing: Noise Knowledge Estimation for Robust Speech Systems", IEEE Trans. Speech & Audio Proc., October 2005.

[4] J. H. L. Hansen, X. Zhang, M. Akbacak, U. Yapanel, B. Pellom, W. Ward, Chapter 2, *DSP in Mobile and Vehicle Systems*, H. Abut, J.H.L. Hansen and K. Takeda (Editors) Springer, 2005.

[5] R. Bakis, S. Schen, P. Gopalakrishnan, R. Gopinath, S. Maes, and L. Polymenakos, "Transcription of Broadcast News - System Robustness Issues and Adaptation Techniques", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 711-714, April 1997.

[6] U. Jain, M. A. Siegler, S. J. Doh, E. Gouvea, J. Huerta, P. J. Moreno, B. Raj, and R. M. Stern, "Recognition of Continuous Broadcast News with Multiple Unknown Speakers and Environments", Proceedings of the ARPA Workshop on Speech Recognition Technology, pp. 61-66, February 1996.

[7] R. Bakis, S. Chen, P. Gopalakrishnan, R. Gopinath, S. Maes, L. Polymenakos, and M. Franz, "Transcription of Broadcast News Shows with the IBM Large Vocabulary Speech Recognition System", Proceedings of DARPA Speech Recognition Workshop, pp. 67-72, February 1997.

[8] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic Segmentation, Classification and Clustering of Broadcast News Audio", Proceedings of DARPA Speech Recognition Workshop, pp. 97-99, February 1997.

[9] J. S. Lim, "Speech Enhancement", Prentice Hall, Englewood Cliffs, NJ, 1983.

[10] J. H. L. Hansen, *Speech Enhancement, Encyclopedia of Electrical and Electronics Engineering*, John Wiley & Sons, vol. 20, pp. 159-175, 1999.

[11] J. G. Fiscus, "A Post Processing System to yield reduced error rates: Recognizer Output Voting Error Reduction (ROVER)", IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 347-54, 1997.

[12] S. Chen and P. S. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion", Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 127-132, February 1998.

[13] B. Zhou and J. H .L. Hansen, "Unsupervised Audio Stream Segmentation and Clustering via the Bayesian Information Criterion", Proc. of Inter. Conf. on Spoken Language Processing ICSLP-2000, vol. 3, pp. 714-717, October 2000.

[14] G. Zhou, J. H. L. Hansen, and J. F. Kaiser, "Nonlinear Feature Based Classification of Speech under Stress", IEEE Trans. on Speech & Audio Processing, vol. 9, no. 2, pp. 201-216, March 2001.

[15] Y. Gong, "Speech Recognition in Noisy Environments: A Survey, Speech Communication, vol. 16, pp. 261-91,1995.

[16] C. J. Leggetter and P. C. Woodland, "Maximum Likelihood Linear Regression for speaker adaptation of continuous density hidden Markov models", Computer Speech and Language, vol. 9, no. 2, pp. 171-185, April, 1995.

[17] M. Gales and S. Young, "Robust Continuous Speech Recognition using Parallel Model Combination", IEEE Transactions on Speech and Audio Processing, vol. 4, pp. 352-359, September 1996.

[18] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech", Journal of the Acoustical Society of America, vol. 87, no. 4, pp. 1738-1752, 1990.

[19] R. Sarikaya and J. H. L. Hansen, "High Resolution Speech Feature Parameterization for Monophone Based Stressed Speech Recognition", IEEE Signal Processing Letters, vol. 7, no. 7, pp. 182-185, July 2000.

[20] R. Sarikaya and J. H. L. Hansen, "Robust detection of Speech Activity in the Presence of Noise", International Conference on Spoken Language Processing, vol. 4, pp. 1455-1458, December 1998.

[21] P. Angkititrakul, J. H. L. Hansen, S. Baghaii, "Cluster-dependent Modeling and Confidence Measure Processing for In-Set/Out-of-Set Speaker Identification", Interspeech-2004/ICSLP-2004: Inter. Conf. Spoken Language Processing, Jeju Island, South Korea, October 2004.

Chapter 11

SPEAKER SOURCE LOCALIZATION USING AUDIO-VISUAL DATA AND ARRAY PROCESSING BASED SPEECH ENHANCEMENT FOR IN-VEHICLE ENVIRONMENTS

Xianxian Zhang[1], John H.L. Hansen[1,3], Kazuya Takeda[2], Toshiki Maeno[2], Kathryn Arehart[3]

[1]*Center for Robust Speech Systems, Dept. of Electrical Engineering, Erik Jonsson School of Engineering & Computer Science, University of Texas at Dallas, Richardson TX, USA (Prevoulsy Robust Speech Processing Group – CSLR, Univ. of Colorado, Boulder, CO, USA)*

[2]*Nagoya University, Nagoya, Japan*

[3]*Dept. of Speech, Language and Hearing Sciences,University of Colorado at Boulder, Boulder CO USA*

Abstract: Interactive systems for in-vehicle applications have as their central goal the primary need to improve driver safety while allowing drivers effective control of vehicle functions, access to on-board or remote information, or safe hands-free human communications. Human-Computer interaction for in-vehicle systems requires effective audio capture, tracking of who is speaking, environmental noise suppression, and robust processing for applications such as route navigation, hands-free mobile communications, and human-to-human communications for hearing impaired subjects. In this chapter, we discuss safety with application for two interactive speech processing frameworks for in-vehicle systems. First, we consider integrating audio-visual processing for detecting the primary speech for a driver using a route navigation system. Integrating both visual and audio content allows us to reject unintended speech to be submitted for speech recognition within the route dialog system. Second, we consider a combined multi-channel array processing scheme based on a combined fixed and adaptive array processing scheme (CFA-BF) with a spectral constrained iterative Auto-LSP and auditory masked GMMSE-AMT-ERB processing for speech enhancement. The combined scheme takes advantage of the strengths offered by array processing methods in noisy environments, as well as speed and efficiency for single channel methods. We evaluate the audio-visual localization scheme for route navigation dialogs and

show improved speech accuracy by up to 40% using the CIAIR in-vehicle data corpus from Nagoya, Japan. For the combined array processing and speech enhancement methods, we demonstrate consistent levels of noise suppression and voice communication quality improvement using a subset of the TIMIT corpus with four real noise sources, with an overall average 26dB increase in SegSNR from the original degraded audio corpus.

Key words: Automatic speech recognition, robustness, microphone array, multi-modal, speech enhancement, environmental sniffing, dialog, mobile, route navigation, in-vehicle

1. INTRODUCTION: HANDS-FREE TELEPHONY AND COMMUNICATIONS IN CARS

Interactive systems for in-vehicle applications have as their central goal the primary need to improve driver safety while allowing drivers effective control of vehicle functions, access to on-board or remote information, or safe hands-free human communications. Over the past thirty years, the complexity of in-vehicle driver instrumentation and user control features has increased significantly. However, the basic requirements necessary for satisfying the requirements to obtain a driver's license have essentially remained constant. In addition, society has placed a value on drivers performing multiple tasks in addition to operating their vehicles including, but not limited to, eating, drinking, talking on their cellular telephones, listening to music or radio, accessing voice-mail or information resources via their cell phone, carrying on conversations with passengers or maintaining order in the vehicle if children are present, etc. Finally, the range of driver skills is also quite broad, with older and younger drivers typically less capable of managing multiple tasks while driving. Recent studies have suggested that the increased complexity of in-vehicle technology, if left unchecked, will increase driver task demands and produce further distractions. Driver distraction is one of the leading causes of automobile accidents from recent U.S. NTSB investigations [1] (see Preface in this textbook). Therefore, to improve safety, there is a need for future in-vehicle systems to adjust to the users' capability, both overall and in real-time during vehicle operation.

Human-computer interaction for in-vehicle information access, human communications, and route navigation systems are challenging problems because of the diverse and changing acoustic environments inside cars [2]. There are many situations where it is important to be able to identify and track which subject inside the vehicle is desired talking, as well as perform

speech processing using multi-microphone arrays and enhancement algorithms to improve the perceived quality of speech. Some sample environments include but not restricted to: in-vehicle hands-free voice communications, mobile phone use in public noisy environments, hearing impaired persons in large classrooms or meeting halls, and others. A number of speech enhancement algorithms have been proposed in the past, and a survey can be found in ([3] - Chap. 8).

In this chapter, we consider two aspects of signal processing for in-vehicle systems with emphasis on improved safety: (i) audio-visual processing for localization of the primary talker for in-vehicle route dialog systems with interest in reducing driver distraction/task stress, and (ii) combine array processing and auditory based speech enhancement to improve communications within the car environment for hearing impaired drivers. In the first area, it is proposed that the integration of video and audio information can significantly improve dialog system performance, since the visual modality is not impacted by acoustic noise. Here, we propose a robust audio-visual integration system for source tracking and speech enhancement for an in-vehicle speech dialog system. The proposed system integrates both audio and visual information to locate the desired speaker source. Using real data collected in car environments, the proposed system can improve desired speech accuracy by up to 40.75% compared with audio data alone.

In the second area, we consider array and speech enhancement processing to improve communications safety for hearing impaired drivers. In the United States, there are not any vehicle operating restrictions for hearing-impaired drivers. However communications between driver, passengers, or in-vehicle route navigation systems may not be effective for hearing-impaired drivers. Speech enhancement for hearing-impaired subjects inside car environments requires FM technology where speech from non-hearing impaired speakers are captured and transmitted via a wireless link directly to a hearing assist device worn by the hearing impaired subject. One way to discuss trade-offs in speech enhancement algorithms in this area is to separate those that are single-channel, dual channel, or multi-channel array based approaches. For single-channel applications, only a single microphone is available. Characterization of noise statistics must be performed during periods of silence between utterances, requiring (i) a stationary or short-time varying assumption of the background noise, and (ii) that the speech and noise are uncorrelated. In this area, we incorporate array processing with single channel speech enhancement methods to suppress noise for in-vehicle applications. In the next section, we consider localization in the car environment.

This chapter is organized as follows. In Sec. 2, we present our proposed in-vehicle audio-visual system. In Sec. 3, we discuss the array speech

enhancement for hearing impaired drivers in vehicle environments. Sec. 4 concludes with a summary and discussion of areas for future work.

2. LOCALIZATION VIA AUDIO-VISUAL PROCESSING

The increased use of mobile telephones and voiced controlled features for human-machine dialog system in cars has created a greater demand for hands-free, in-car installations. Many countries now restrict handheld cellular technology while operating a vehicle. As such, there is a greater need to have reliable voice capture within automobile environments.

However, the distance between a hands-free car microphone and the speaker will cause a severe loss in speech quality due to changing acoustic environments. Therefore, the topic of capturing clean and distortion-free speech under distant talker conditions in noisy car environments has attracted much attention. Microphone array processing and beamforming is one promising area which can yield effective performance. Currently, most beamforming algorithms have to incorporate speaker/source localization techniques in order to enhance the desired speech and suppress interference [4,5,6].

Here, speaker localization is the ability to estimate the position of a speaker in the car, and involves the following:

Complex in-vehicle noise situations will severely degrade performance of speaker localization techniques.

Speaker localization techniques cannot distinguish between desired and undesired speech if both speech sources are from the same direction.

In the vehicular environment, the desired speech for a navigation system is assumed to be the one from the driver, while the undesired speech includes both the passengers and a portion of driver's (e.g., the driver murmurs while looking up or down, the driver laughs and chats with other people inside car, etc.). One way to address this problem is to integrate visual based object localization techniques.

Audio-visual (A-V) speaker localization has recently received significant interest [7,8,10] mainly because the visual modality is not affected by varying acoustic noise and sound localization is unaffected by changing light conditions. However, there are situations where the integration of video information can significantly improve in-vehicle human-machine dialog system performance. For example, determining the movement of the driver's mouth, body, and head position can impact how a dialog system should respond. If the driver's mouth does not move while speech is detected from the driver's position, then most likely the passenger who sits behind the

driver is talking. If the driver asks a question while facing forward, then we can expect the request is being directed towards the in-vehicle dialog system. If the driver is turned towards individuals sitting in the backseat, then the question is most likely directed at someone in the car (e.g., "Where did you say you wanted to eat?"). For such a case, it would not be appropriate to submit such a request to the dialog system. In this area, we discuss the development of an audio-visual system for in-vehicle localization which was originally developed in [10]. Evaluations are based on data collected from the automobile collection platform at the Center for Integrated Acoustic Information Research (CIAIR) [11], Nagoya University, Japan.

2.1 Audio-Visual Integration System

Figure 11-1 illustrates the proposed audio-visual integration system which includes the following four stages: audio-visual data synchronization, speaker localization using audio data, face tracking using visual data, and speech enhancement and noise/interfering speech suppression using a constrained switched adaptive beamformer [4].

2.1.1 A-V Data Synchronization: Since sampling rates of the audio and visual signals are different, the proportion of the number of the sampled audio data to that of the visual data in general is a fractional frame number. In our case, the CIAIR database from Nagoya Univ. [11] uses a 16.0 kHz speech sampling rate with a video data rate of 30 frames per second. After synchronization, we keep the temporal mismatch error between the audio and visual data at less than 0.033 sec. This mismatch level is acceptable since visual data is only used for speaker localization and activation.

2.1.2 Source Tracking Using Audio: In tracking signal energy, we employ the Teager Energy Operator (TEO) [19,20]. TEO based process has recently been extended to address detection and assessment of speech under stress and emotion [21]. Here, we first use the TEO criterion to decide the speech activity for the audio data, and then apply the adaptive LMS filter technique to locate the current position of the speech source. Further details are discussed in [8]. We define the angular direction where the driver's head is position as shown in Figure 11-1. Here, we have quantized the angular position in steps of 2.8 degrees.

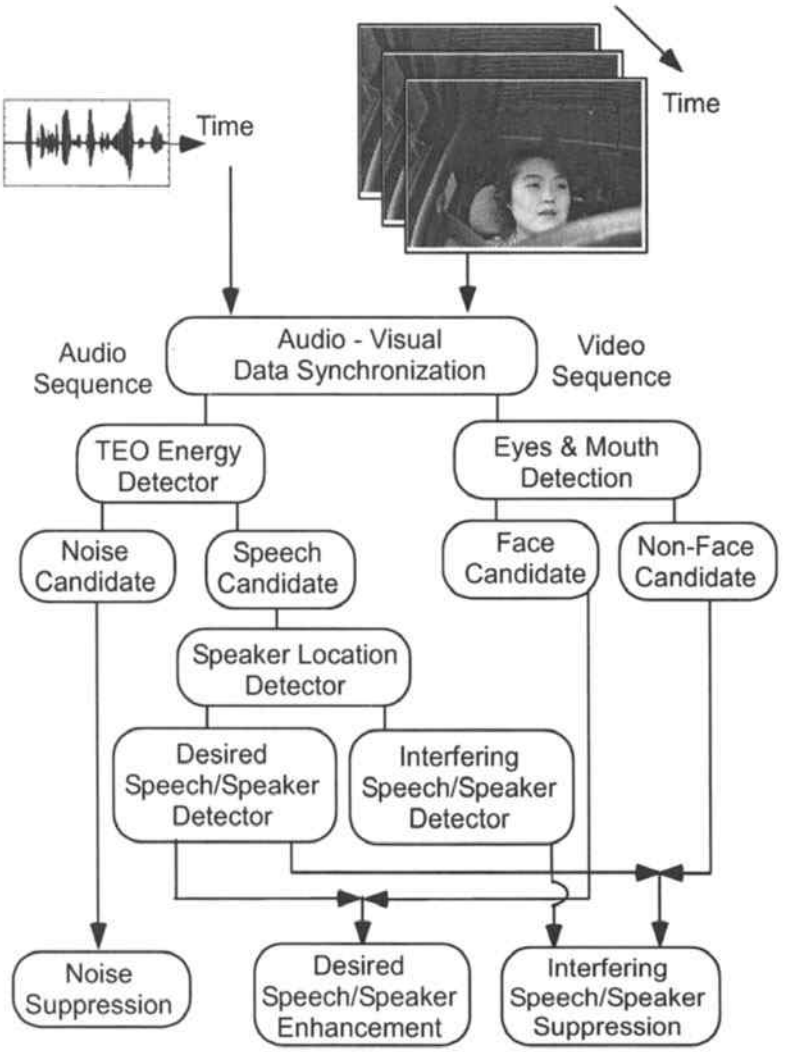


Figure 11-1. Schematic diagram of the proposed audio-visual integration system.

2.1.3 Face Tracking Using Visual Data: The function of this processing stage is to detect interfering speech which cannot be identified by sound localization techniques alone. We apply basic eye and mouth detection and tracking techniques in this processing stage. From our observation and experiments using the CIAIR in-vehicle corpus, we found that most of the interfering speech versus that from the driver occurs in the following situations:

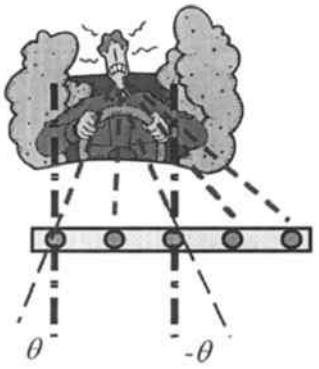
Case 1: The passenger talks and the driver listens: Under this situation, the driver's lips will not move often;

Case 2: The driver murmurs while looking up or down, which causes part of his/her face to be obscured by the steering wheel;

Case 3: The driver laughs or coughs while covering his/her mouth with their hands;

Case 4: The driver chats with the interfering person while he/she is driving. Under this situation, the driver will likely shift his/her head or body slightly towards the interfering person, which makes a portion of the face features disappear.

Figure 11-2 shows examples where there is speech interference (i.e., speech not always intended for route navigation). In the formulated audio-visual integration system, we use template based eyes and mouth detection software to detect face features. We also track the distance between the eyes



position number	corresponding θ
0	0
1	2.8°
2	5.6°
3	8.4°
4	11.2°
5	14°
6	16.8°
7	19.6°
8	22.4°
-1	-2.8°
-2	-5.6°
unknown	unknown

Figure 11-2. Driver orientation with respect to 5 channel microphone array. Position number corresponds to angular shift of driver's head; positive values mean head shifts to the drivers right; negative values mean shifts to the drivers left (Note: -3,-4,-5, etc. correspond to further angle positioning to the left of the driver).

and changing mouth shape across frames. For example, if a driver's mouth shape does not change within a certain period, the current speech most likely comes from the passenger (i.e., Case 1) {Figure 11-3a}; if the distance between driver's eyes is smaller than a certain value for a time period, then he/she likely has shifted their head backwards (i.e., Case 4) {Figure 11-3f}; if part of face features, such as the mouth, cannot be detected, then most likely the driver is under situations described in Case 2 and 3 {Figure 11-3e}.

2.1.4 Enhancement and Interfering Speech/Noise Suppression: Once we detect the nature of the current signal, we propose to use the constrained switched adaptive beamforming algorithm (CSA-BF) [2] to enhance the desired speech and suppress background noise and interfering speech.

Figure 11-3a

Figure 11-3b

Figure 11-3c

Figure 11-3d

Figure 11-3e

Figure 11-3f

Figure 11-3. (a-d): Face is detected as "existing" and source comes from quantized angular locations 1, 0, -3, and -5 respectfully (see Figure. 11-2 for angular definitions). (e-f): face is detected as "not existing" and source is from direction 0 (driver laughing) and -5 (interference from passenger talking with the driver).

2.2 Evaluation

Figure 11-4 shows how visual information helps to detect the interfering speech or non-dialog directed speech. Here, it is straightforward to determine when speech activity occurs, but a greater challenge is to say when speech is directed towards the microphone array based in-vehicle navigation system. For example, the signal during the period from frame count 150 to 200 corresponds to when the driver is laughing. Here, the averaged Teager energy (TEO) is high enough to pass the speech threshold, and sound localization results also confirm that the speech comes from position number 0, (i.e., the driver is talking and facing forward). Therefore, if only audio information is used, the speech during this period will be identified as desired speech. However, from the results of face feature detection, we find that the driver's mouth cannot be detected since it is covered by the driver's hand, and therefore this segment is correctly labeled as interfering speech. Similarly, when the driver is talking with the

passenger during frame count 400 to 500, our face tracking algorithm is also able to classify this speech as interfering speech, since the driver shifts her head backward frequently while chatting with the passenger, and the detected distance between the eyes is shorter than that while facing forward. This speech also cannot be identified as undesired by audio data only.

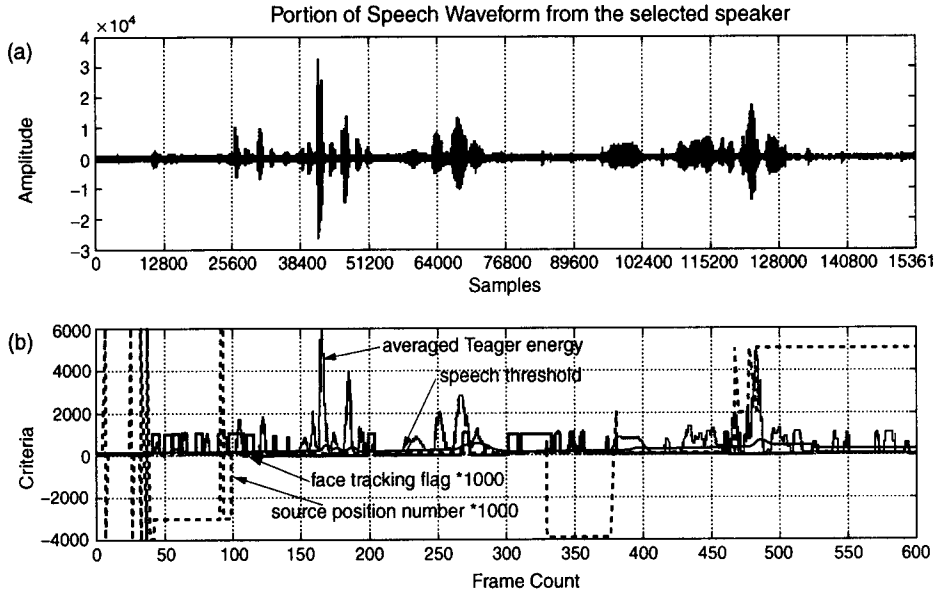


Figure 11-4. Audio-visual tracking results for a selected speaker using actual in-vehicle audio-visual data from the CIAIR corpus.

Table 11-1. Performance of Detected Route Dialog Directed Speech using (i) manual labeling, (ii) audio processing alone, and (iii) audio and visual processing (note: this particular audio-visual stream consists of 321.5 sec of data, of which there is 107.335 sec of speech activity).

Manually Computed Periods (in secs)		Detected Periods (in secs) Using Audio Data		Detected Periods (in secs) Using Audio-Visual Data	
Speech	Desired speech	Speech	Desired speech	Speech	Desired speech
107.335	47.173	128.392	95.978	128.392	57.456

Table 11-1 shows the accumulated speech and desired speech activity periods under different experimental situations. We can see that by using visual data processing in addition to the TEO criterion with the LMS filter, the accuracy of the desired speech detection is improved 40.75%, (i.e., a reduction in the desired speech duration from 96 sec to 57 sec, approaching the goal of 47 sec). While this improvement is important (i.e., eliminating

39.412 sec. of undesired speech), there is still approximately 10 seconds of interfering speech that remains.

2.3 SUMMARY: Interfering Speech/Speaker Cancellation via Audio-Visual Data

For the system from Figure 11-1, the interfering speech cancellation is possible with/without face tracking results as one of the constraints for the constrained switched adaptive beamforming (CSA-BF). From these results, we can make the following observations:

Employing the proposed audio-visual integration system can improve the accuracy of the desired source tracking by up to 40.75% (i.e., we can remove non-desired speaker speech prior to ASR for the dialog system).

The proposed system with better source tracking using Audio-Visual also improves interfering speech cancellation.

3. SPEECH ENHANCEMENT: ARRAY + SINGLE CHANNEL PROCESSING SCHEMES

In this section, we consider speech/array processing for hearing impaired subjects for in-vehicle environments. Background car noise and competing speakers interference represents challenges for hearing-impaired subjects in car environments. For the application for hearing impaired subjects in vehicle, we first present a data collection experiment for a proposed FM wireless transmission scenario using a 5-channel microphone array in the car, and followed by several alternative speech enhancement algorithms. After formulating 6 different processing methods, we evaluate the performance using SegSNR improvement with data recorded in a moving car environment. Among the 6 processing configurations, the combined fixed/adaptive beamforming (CFA-BF) obtains the highest level of SegSNR improvement by up to 2.65 dB. An earlier version of this work was presented in [12], and here we discuss the framework and further results in detail.

To motivate the proposed method, we consider a previous proposed combined fixed/adaptive beamforming algorithm (CFA-BF) [11] for a TIMIT sentence degraded by Flat Channel Communication Noise (FLN). We use the same microphone array set up, and found that this method can improve SegSNR (Signal-to-Noise Ratio) by up to 11.75 dB. Next, we also applied a recently proposed GMMSE-AMT-ERB algorithm (GAE) [15] that uses an auditory masked threshold with equal rectangular bandwidth filters,

and an earlier spectral constrained iterative speech enhancement algorithm Auto-LSP [16] on the same noisy data, and found that the SegSNR improvements are 16 dB and 20.5 dB respectively. However, these algorithms cannot entirely suppress the FLN noise. Fig. 11-5 shows the spectrogram of the original degraded speech, and enhanced speech by CFA, GAE, and Auto-LSP respectively [16]. Our original objective of choosing FLN noise was to focus on the design of an algorithm that can obtain the best performance under this stationary noise condition, and then to extend it to more complex noise environments. From the above experimental results, we see that CFA is able to suppress high frequency noise, GAE suppresses noise uniformly, and Auto-LSP suppresses noise efficiently across the entire band, but there is still some residual noise in the high frequency region.

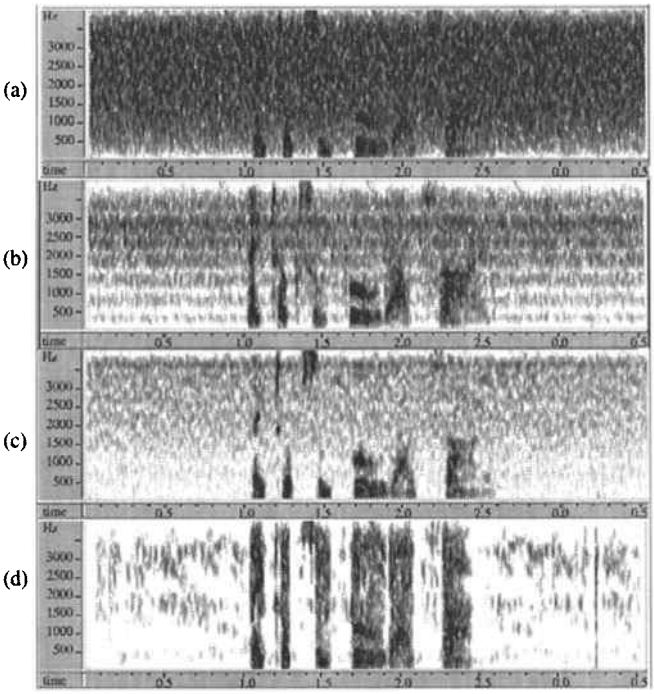


Figure 11-5. Spectrogram of Speech Data with: (a). Original FLN degraded noisy speech; (b). CFA Enhanced speech; (c). GMMSE-AMT-ERB Enhanced speech; (d). Auto-LSP Enhanced speech.

3.1 Overall Algorithm Description

In our algorithm, we first apply combined fixed/adaptive beamforming (CFA-BF) for front-end processing to obtain a first stage enhanced speech signal by suppressing high frequency noise as well as generating a

corresponding residual noise. Secondly, according to the nature of the noise and the angle between the direction of speech and interference, we select a back-end processing method from 3 possible spectral based speech enhancement algorithms to suppress residual noises (i.e. enhancement scheme #1, #2 or #3). Figure 11-6 summarizes an overall description of the proposed algorithm.

3.1.1 Detailed Algorithm Design

3.1.1.1 Front-end processing

The block diagram of the structure of the proposed algorithm is shown in Figure 11-7. We know that most of adaptive beamforming algorithms will select one of the microphones as the primary microphone, and build an adaptive filter between it and each of the other microphones. These filters compensate for the different transfer functions between the speaker and the microphone array. Therefore, there are two kinds of outputs from the adaptive beamforming algorithm: namely the enhanced speech $d(n)$ and noise signal $e_i(n)$. Here, when we use the combined fixed/adaptive beamforming algorithm (CFA-BF) [7], we choose microphone 0 as the primary microphone, therefore, the enhanced speech $d(n)$ and noise signal $e_i(n)$ are given as in Eqn. (1) and (2):

$$d(n) = \frac{1}{N}\sum_{i=0}^{N-1} w_i^T(n)x_i(n) \tag{1}$$

$$e(n) = w_0^T(n)x_0(n) - w_i^T x_i(n) \tag{2}$$

where, N is the total number of microphones, x_i is the i^{th} microphone input signal with $i = 0, 1, ..., N-1$. Compared with the original noisy speech, the enhanced speech $d(n)$ suppresses noise mainly in the high-frequency band, and the corresponding noise outputs $e_i(n)$ are the residual noises that are synchronous with $d(n)$ in time, but asynchronous with $d(n)$ in phase.

Let: ϕ be the angle between the speech source and the axis of the microphone array, ψ be the angle between the interference and the axis of the microphone array, θ_1 be the lower bound of the angle threshold, θ_2 be the upper bound of the angle threshold; then,

1. if $|\phi - \psi| \geq \theta_1$, then go to Step 4;
2. if $|\phi - \psi| \leq \theta_2$, then select scheme #2;
3. if $\theta_1 < |\phi - \psi| < \theta_2$, then we are between performance bounds for the methods, so we can randomly select one of the schemes to use, or employ other criteria to select the proper scheme to use;
4. if the current noise has strong low frequency content, then select scheme #2; else select scheme #1.

Here, both the angle and threshold are decided by the geometry of the microphone array, the distance from the sources to the array, and the nature of the interference.

Figure 11-6. Formal description of the proposed algorithm.

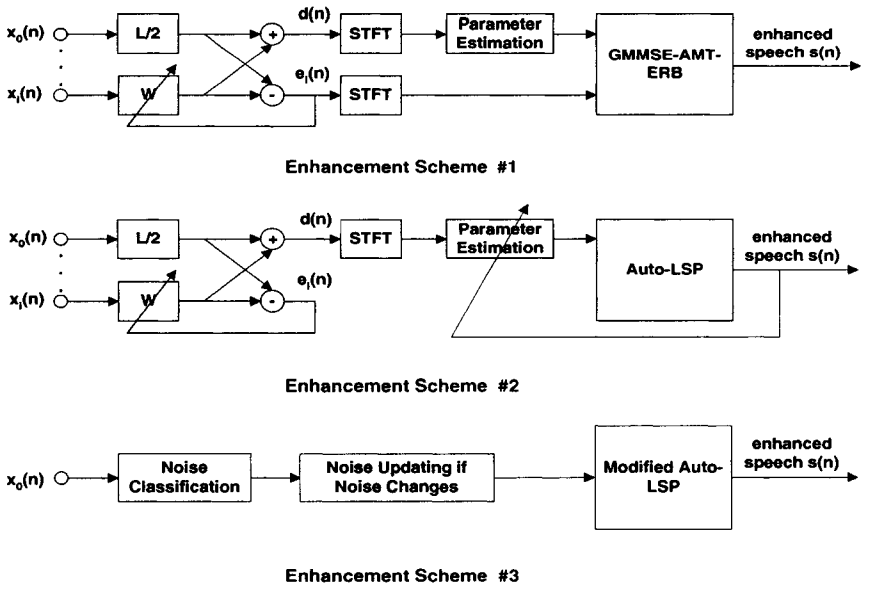


Figure 11-7. Block Diagram of the Proposed Algorithm.

3.1.1.2 Back-end processing

For the back-end processing, we propose 3 possible enhancement schemes, which are classified into 2 categories:

Category 1: includes scheme #1 and #2. Both enhancement schemes use the outputs of front-end processing as the input for back-end processing;

Category 2: includes scheme #3 only. This scheme uses the microphone array as a tool to classify the current noise. If the current noise changes, noise updating will be performed to provide current noise estimation for back-end processing. The input of the back-end processing here will be the original input signal of the primary microphone.

In scheme #1, we adapt a modified GMMSE-AMT-ERB (mGAE), which builds on the original MMSE method [13]. The original GAE is proposed in [14] and assumes that the speech is degraded with additive noise and the speech and noise segments are uncorrelated as in (3):

$$y(n) = x(n) + n(n) \tag{3}$$

The short term power spectrum is calculated by applying a Hamming window to a frame of speech. Under this assumed model, one can obtain a family of MMSE speech spectral estimators as,

$$\hat{X}_p = (E\{X_p^a \mid Y_p\})^{1/a} \tag{4}$$

Here, let P_{nk} be the noise power spectrum for the k^{th} subband, and P_{yk} be the noisy speech power spectrum for the k^{th} subband. The values of P_{nk} and P_{yk} are calculated as follows,

$$P_{nk}[n] = \gamma P_{nk}[n-1] + \frac{1-\gamma}{1-\beta}(P_{yk}[n] - \beta P_{yk}[n-1]) \tag{5}$$

$$P_{yk}[n] = \alpha P_{yk}[n-1] + (1-\alpha)(Y_k \mid [n]\mid^2 \tag{6}$$

In our implementation, the first ten frames of noisy speech, which consists of only noise, is taken as the estimation of the noise for the entire noisy speech sentence. This assumption is valid if the noise does not change. However, once the noise spectrum changes, enhancement performance will decrease, resulting in either under or over noise suppression. Therefore, in the modified GAE (mGAE) algorithm, we use the residual noise $e_i(n)$ that is generated by beamform front-end processing instead of the noise spectrum estimation of GAE in scheme #1. Under the proposed model, Eqn (5) now becomes,

$$P_{nk}[n] = \sum_{i=1}^{N-1} \lambda_i P_{e_i k}[n] \tag{7}$$

$$P_{e_i k}[n] = | e_i[n] |^2 \tag{8}$$

where λ_i is a scaling factor, and we use $\lambda_i = 1/N$ for all $i = 1, ..., N-1$.

In scheme #2, we use the enhance speech $d(n)$ as an input of the Auto-LSP algorithm to remove the residue noise. This algorithm is discussed in more detail in [1] and [15].

Scheme #3 is selected only when the speech source and interference are very close to each other. Since beamforming algorithms (delay-and-sum beamforming or adaptive beamforming) obtain the enhanced signal by selecting the appropriate delays (fixed or adaptive) between each microphone and summing the delayed signals in phase for direction angle θ, we will have destructive interference for signals arriving from other angles. Fortunately, we can obtain a good noise estimate using single channel processing under this situation. Once a noise change is detected, noise spectrum updating is performed. We do not update the noise spectrum frame by frame, since we believe this will increase speech distortion. With the aid of a noise classification stage, a modified Auto-LSP algorithm (mAutoLSP) is used here as the back-end processing solution. The difference between mAuto-LSP and Auto-LSP is the presence (e.g. with/without) of the noise classification stage.

3.1.1.3 Performance Evaluation

(i.) Experimental Database & Setup

In order to evaluate the performance of the proposed algorithm, we select 10 sentences from the TIMIT database, and degrade these sentences with four different noise sources: (i) White Gaussian Noise (AWG), (ii) Flat Channel Communication Noise (FLN), (iii) Large Crowd Room Noise (LCR), and (iv) Automobile Highway Noise (HWY). The sample frequency of both the sentences and noises is 8k Hz. The noise level is adjusted to be an overall average 5dB SNR. For evaluations, we use the Segmental Signal-to-Noise Ratio (SegSNR) measure [16], which represents a noise reduction criterion for voice communications.

(ii.) Experiment Results

Figure 11-8 illustrates average SegSNR improvement using sentences degraded with FLN noise. Table 11-2 show the Segmental SNR measure for

the degraded speech with 4 different noises and enhanced speech by 5 different schemes.

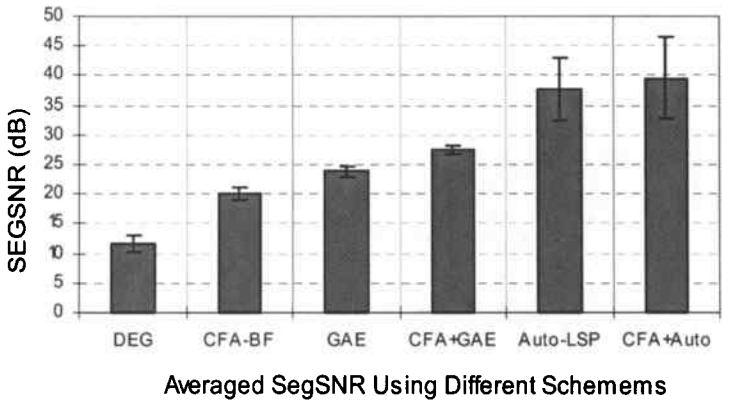


Figure 11-8. SegSNR Results for Degraded and Enhanced Speech.

From these results, we can see that employing the proposed algorithm (array processing combined with either the psycho-acoustically motivated GMMSE-AMT-ERB or speech based spectral constrained Auto-LSP), increases SegSNR significantly compared with any one individually. The SegSNR improvement is up to 26 dB over the original degraded corpus set. Finally, an informal listener test evaluation confirmed the level of noise suppression and quality improvement for the proposed method.

Table 11-2. Averaged Segmental SNR (dB) for Different Schemes.

NOISE	**DEG**	**CFA-BF**	**GAE**	**CFA-BF + GAE**	**Auto-LSP**	**CFA + Auto-LSP**
FLN (5dB)	11.55	20.1	23.775	27.575	37.55	39.525
LCR (5dB)	13.775	21.35	23.875	29.825	27.125	37.525
HWY (5dB)	12.1	13.35	18.975	16.225	36.925	39.4
AWN (5dB)	8.15	14.175	18.275	19.975	32.525	32.5
Avg. across noises	11.39	17.24	21.23	23.4	33.53	37.24

4. DISCUSSION

In this chapter, we have considered two interactive speech processing frameworks for in-vehicle systems. First, we considered integrating audio-visual processing for localization the primary speech for a driver using a route navigation system. Integrating both visual and audio content allows us to reject unintended speech to be submitted for speech recognition within the route dialog system (i.e., a 40.75% improvement). Next, we considered a combined multi-channel array processing scheme based on CFA with a spectral constrained iterative Auto-LSP and auditory masked GMMSE-AMT-ERB processing for speech enhancement. The combined scheme takes advantage of the strengths offered by array processing methods in noisy environments, as well as speed and efficiency for single channel methods. We evaluated the enhancement methods on a section of the TIMIT corpus using four different actual noise conditions. We demonstrated a consistent level of noise suppression and voice communication quality improvement using the proposed method as reflected by an overall average 26dB increase in SegSNR from the original degraded audio corpus. In the future, we plan to study algorithm sensitivity to more time varying noise sources as well as reverberant environments. These contributions suggest that improvements for interactive systems for in-vehicle systems such as multi-sensor based schemes and assist frameworks for hearing-impaired users can expand the use of in-vehicle route navigation systems as well as hands-free and human communication devices for cars.

REFERENCES

[1] B. A. Magladry, Director of the Office of Highway Safety, U.S. National Transportation Safety Board (N.T.S.B.) – personnel communication during "2005 Workshop on DSP for In-Vechile & Mobile Systems," Sesembra, Portugal, Sept. 2005 (http://dspincars.sdsu.edu).

[2] J. H.L. Hansen, X.X. Zhang, M. Akbacak, U.H. Yapanel, B. Pellom, W. Ward, P. Angkititrakul, "CU-MOVE: Advanced In-Vehicle Speech Systems for Route Navigation," Chapter 2 in *DSP for In-Vehicle and Mobile Systems*, (Abut, Hansen, Takeda, Ed.s), Springer-Verlag Publishers, 2004.

[3] J. R., Deller, J. H.L. Hansen, and J.G. Proakis, *Discrete-Time Processing of Speech Signals*, Ch. 8, Speech Enhancement, (2nd Edition), IEEE Press, New York, NY, 2000.

[4] X. Zhang, J.H.L. Hansen, "CSA-BF: A Constrained Switched Adaptive Beamformer for Speech Enhancement and Recognition in Real Car Environments," IEEE Trans. Speech & Audio Proc., vol. 11, no. 6, pp. 733-745, Nov. 2003.

[5] D. von Compernolle, "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," IEEE ICASSP-90, April 1990.

[6] E. Visser, M. Otsuka, T.W. Lee, "A spatio-temporal speech enhancement scheme for robust speech recognition," ICSLP-2002, Denver, CO, Sept. 2002.

[7] C. Neti, G. Potamianos, et. al, "Audio-Visual Speech Recognition," Final Workshop Report, Johns Hopkins Univ., 2000.
[8] M. Beal, N. Jojic, H. Attias, "A self-calibrating algorithm for speaker tracking based on audio-visual statistical models, IEEE ICASSP-02.
[9] X.X. Zhang, J. H.L. Hansen, "CFA-BF: A Novel Combined Fixed/Adaptive Beamformer for Robust Speech Recognition in Real Car Environments," Eurospeech-2003, pp. 1289-92, Sept. 2003.
[10] X.X. Zhang, K. Takeda, J.H.L. Hansen, T. Maeno, "Audio-Visual Integration for Hands-Free Voice Interaction in Automobile Route Navigation," ICA-2004: International Congress on Acoustic, vol. 4, pp. 2821-2824, Kyoto, Japan, April 2004.
[11] http://www.ciair.coe.nagoya-u.ac.jp/
[12] X.X. Zhang, J. H.L. Hansen, K. Arehart, "Speech Enhancement based on a Combined Multi-Channel Array with Constrained Iterative and Auditory Masked Processing," IEEE ICASSP-2004, vol. 1, pp. 229-232, Montreal, Canada, May 2004
[13] A. Natarajan, J. H.L. Hansen, K. Arehart, and J. Rossi-Katz, "Perceptual Based Speech Enhancement for Normal-Hearing & Hearing-Impaired Individuals", Interspeech/Eurospeech-'2003, pp.1425-1428 Geneva, Switzerland, 2003.
[14] J. H.L. Hansen, M. Clements, "Constrained iterative speech enhancement with application to speech recognition", IEEE Trans. Signal Processing, vol. 39, no. 4, April, 1991.
[15] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator", IEEE. Trans. on Acoustics, Speech, and Signal Processing, vol. 32, 1984.
[16] Y. Ephraim, "Statistical-Model-Based Speech Enhancement Systems," Proceeding of the IEEE, **80**(10):1526-1555, 1992.
[17] J. H.L. Hansen, S. Nandkumar, "Robust Estimation of Speech in Noisy Backgrounds Based on Aspects of the Auditory Process," Journal Acoustical Society of America, vol. 97, no. 6, June 1995.
[18] http://www.nist/gov/
[19] H. M. Teager, "Some observations on oral air flow during phonation," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-28, no. 5, pp. 599-601, Oct. 1980.
[20] H. M. Teager and S.M. Teager, "Evidence for nonlinear production mechanisms in the vocal tract," in Speech Production and Speech Modeling, Norwell, MA, Kluwer, vol. 55, pp. 241-261, 1989.
[21] G. Zhou, J. H.L. Hansen, and J.F. Kaiser, "Nonlinear Feature Based Classification of Speech under Stress," IEEE Transactions on Speech & Audio Processing, vol. 9, no. 2, pp. 201-216, March 2001.

Chapter 12

ESTIMATION OF ACTIVE SPEAKER'S DIRECTION USING PARTICLE FILTERS FOR IN-VEHICLE ENVIRONMENT

Mitsunori Mizumachi and Katsuyuki Niyada
Kyushu Institute of Technology, 1-1 Sensui-cho, Tobata-ku, Kitakyushu-shi, Fukuoka 804-8550, Japan

Abstract: Building in-vehicle human-machine interfaces, information on speaker's direction is helpful for achieving robust speech recognition and video camera control applications. This chapter presents two-step particle filtering approach in a spectro-spatial domain for achieving robust Direction-Of-Arrival (DOA) estimation in noisy environments such as in-vehicle conditions. The particle filter is applied to track the movement of an active speaker while speaking, and the two-step filtering aims at combining the advantages of both traditional cross-correlation (CC) and generalized cross-correlation (GCC) methods.

Key words: DOA estimation; cross-correlation; generalized cross-correlation; particle filter; spectro-spatial domain; noisy environment; in-vehicle situation; DOA histogram.

1. INTRODUCTION

Speech communication is an indispensable means for achieving human-machine interaction with minimal distraction for in-vehicle applications. To achieve robust human-machine speech communication, heavily noisy and echo-prone speech is needed to be enhanced by beamforming techniques with signal directions under adverse conditions[1]. Then, Direction-Of-Arrivals (DOAs) of target speech signals have to be estimated accurately in noisy conditions.

Cross-correlation (CC) between two received signals has been widely used for finding a DOA due to its simplicity and efficiency. Among cross-correlation-based approaches, generalized cross-correlation (GCC) method is the most popular technique[2]. The GCC method uses filtered signals, such as whitened-signals, for calculating cross-correlation functions, because wide-band signals generally yield a sharp main-lobe at the true DOA in a correlation function. It is obvious that the sharp peak gives an accurate DOA, and is robust against acoustic interferences. Consider the situation that a narrow-band target signal and one of many wide-band interference signals are observed simultaneously. In this case, the GCC method could not find the true DOA of the target signal even if signal-to-noise ratio (SNR) is high enough.

This chapter presents a robust cross-correlation-based DOA finder by particle filtering in noisy environments. Conventional Kalman filtering approach can tackle this problem if the behavior of both target and noise sources can be modeled by conventional linear statistical models. From the viewpoint of DOA estimation, occlusion, mutations, and rapid changes on signal directions should be considered in the real world scenarios, in particular, when an active sound source disappears or another sound source is born. Furthermore, spectra of the target signals dynamically change time by time in non-stationary noisy conditions. The particle filter can be applied to estimate the state space with relax constraints on stochastic characteristics of sound sources[3].

Target tracking fits well to the class of the suitable problems to be solved by particle filters. Because the voluntary movement of a target can be modeled by Markov process under non-Gaussian, non-stationary, practical noise conditions[4]. The particle filter has been already applied in target tracking for audio, video, and radar applications[3,5,6,7]. Vermaak *et al.* introduced a particle filter for tracking a moving sound source in reverberant environment[8,9]. Ward and Williamson proposed a particle filter beamforming technique for sound source localization[10]. This beamformer-based scheme is attractive for speech enhancement, because it does not require intermediate DOA estimates before beamforming as it is commonly done. Asoh *et al.* proposed a technique on audio and video information fusion by particle filtering for multiple speaker tracking [11]. To simplify the problem, the inherent observation model consisted of two kinds of extracted features, that is, DOAs estimated by the MUSIC algorithm [12] and speakers' positions estimated from video [11]. However, a number of microphones are needed in beamformer-based approaches and the MUSIC algorithm requires *a prior* knowledge on the number of sound sources.

In this study, our observation model consists of non-parametric cross-correlation-based spectro-spatial functions, that is, a set of sub-band cross-correlation functions. Sub-band CC and GCC functions determine the weights for particle filtering in spectral and spatial domains, respectively. Strictly speaking, likelihood is given by half-wave rectified sub-band CC and GCC functions in the spectro-spatial domain. The two-step particle filtering framework is introduced to fuse the emphasized advantages of both traditional CC and GCC methods. We should expect that the proposed method would yield a smooth DOA trajectory, because a frame-based DOA estimate is modeled as a state in the underlying Markov process.

This chapter is organized as follows. In Section 2, we formulate a signal model for DOA estimation, and review traditional CC and GCC methods. In Section 3, a novel DOA estimation approach is introduced to fuse respective advantages of the CC and GCC methods by two-step particle filtering. In Section 4, experimental results are presented to examine the performance of the proposed method compared with the conventional CC and GCC methods. The experiments were performed using target speech recorded in a room with computer-generated white Gaussian noises. Both single source and multiple source conditions are considered. Conclusion is given in Section 5.

1. DOA ESTIMATION BY CROSS-CORRELATION

1.1 Signal Model

Let us assume that a target signal $s(t)$ is received by two spatially-separated microphones M_i and M_j. The signals, $x_i(t)$ and $x_j(t)$, which are received by the microphones, M_i and M_j, respectively, can be simply modeled as follows.

$$\begin{aligned} x_i(t) &= h_i(t) * s(t) + n_i(t), \\ x_j(t) &= h_j(t) * s(t-\tau) + n_j(t), \end{aligned} \tag{1}$$

where $h_i(t)$ represents an impulse response between the sound source and the i^{th} microphone, $n_i(t)$ is a noise signal, which is a mixture of a measuring noise signal and acoustic interferences coming from non-target sound sources, and τ is the time lag when the signal $s(t)$ arrives at the microphones. For far-field sound sources, we assume $h_i(t) \approx h_j(t)$ and difference in phase is more critical for DOA estimation than that the amplitude.

1.2 Cross-Correlation-Based DOA Estimation

In the DOA estimation the cross-correlation-based method is the most frequently used approach. In particular, the GCC method is widely used due to its simplicity and robustness against noise[2]. The GCC function $R_{x_i x_j}(\tau)$ is defined as follows.

$$\begin{aligned} R_{x_i x_j}(\tau) &= \int_{-\infty}^{\infty} \Psi(f) r(\tau, f) df, \\ r(\tau, f) &= X_i(f) X_j^*(f) e^{jf\tau}, \end{aligned} \tag{2}$$

where $X_i(f)$ and $X_j(f)$ are the short-term Fourier transforms (STFT) of $x_i(t)$ and $x_j(t)$, and "*" denotes the complex conjugate. It is also customary to employ a Hanning window to shape the signal. The generalizing operator $\Psi(f)$ is introduced to achieve robust DOA estimation in adverse conditions. If the operator $\Psi(f)$ takes a constant value over the whole frequency, $R_{x_i x_j}(\tau)$ becomes a conventional CC function. However, in this study we employ the WCC method, which uses the GCC function with the whitening operator $\Psi(f)$ defined as

$$\Psi(f) = \frac{1}{|X_i(f)||X_j(f)|}. \tag{3}$$

DOA is given directly from the time lag τ with the peak in a (generalized) cross-correlation function.

2. ROBUST DOA ESTIMATION BY TWO-STEP PARTICLE FILTERING

2.1 Motivation

WCC method has the advantage of accuracy with sharper main-lobe compared with a conventional CC method[2]. When the bandwidth of an acoustic interference is wider than that of the target signal, WCC function is not superior to CC function in finding the DOA of the target signal. This is because WCC function gives a DOA in each frequency irrelevant to energy

distribution, and a global DOA is determined by a vote over DOA estimates. Therefore, band-width dominates the final decision of the global DOA. In the proposed approach, however, we focus at benefiting from both the CC and WCC methods using a two-step particle filtering. Generally, CC function gives rough DOA estimates due to its blunt main-lobe. We expect that the rough estimates by the CC method contribute to finding the true DOAs by the WCC method.

2.2 Problem Statement

Particle filter is flexible on modeling a system compared with other Bayesian filters such as traditional Kalman filters and extended Wiener filters, because it does not require any linearity or Gaussianity on the model. Temporal trajectory of the DOA is modeled by a state-space model, and it is estimated through a state estimation procedure.

Given that the k^{th} frame sampled segment is denoted as $x_{i,k} \equiv x_i(t_k : t_{k+1})$ and $x_{j,k} \equiv x_j(t_k : t_{k+1})$ with the observed continuous signals in time between t_k and t_{k+1}. State estimation is recursively done with the state $\mathbf{z} \equiv (\tau, f)$ on the spectro-spatial state space, which is specified by frequency τ and time lag f. Posterior distribution $p(\mathbf{z}_{1:k} \mid x_{1:k})$ is estimated with sampled signals up to the k^{th} frame, $x_{1:k}$, by particle filtering in Section 3.3.

System model:

Time evolution of the state $\mathbf{z}_k$, which consists of τ_k and f_k, is assumed smoothly changed, and is modeled as

$$\begin{aligned} \tau_k &= \tau_{k-1} + v_k^{(\tau)}, \\ f_k &= f_{k-1} + v_k^{(f)}, \end{aligned} \tag{4}$$

where $v_k^{(\tau)} \sim N(0, \sigma_\tau^2)$ and $v_k^{(f)} \sim N(0, \sigma_f^2)$ are system noises.

Observation model:

CC and WCC functions are calculated from the sampled observed signals, $x_k = (x_{i,k}, x_{j,k})$, through their STFTs X_k.

$$R_{x_k}^{(\Theta)}(\tau) = \sum_{f=F_1,F_2,\cdots} \Psi_k^{(\Theta)}(f) r_k(\tau, f), \tag{5}$$

where $R_{x_k}^{(\Theta)}(\tau)$ corresponds to either CC or WCC function depending on $\Theta = \{CC, WCC\}$. Sub-band CC and WCC functions, in the b-th frequency bin, are obtained as follows.

$$R_{x_k}^{(\Theta)}(\tau, b) = \sum_{f=F_b-\Delta}^{F_b+\Delta} \Psi_k^{(\Theta)}(f) r_k(\tau, f) \tag{6}$$

Then, we regard normalized sub-band CC and WCC functions as likelihood such that

$$p^{(\Theta)}(x_k \mid \mathbf{z}_k) \propto R_{x_k}^{+(\Theta)}(\tau, f), \tag{7}$$

where $R_{x_k}^{+}(\tau, f)$ means the half-wave rectified correlation function at the k-th frame.

State estimation:

State estimation is conducted as follows.

$$p(\mathbf{z}_{1:k} \mid x_{1:k}) \propto p(\mathbf{z}_{1:k-1} \mid x_{1:k-1}) \frac{p(\mathbf{z}_k \mid \mathbf{z}_{k-1}) p(x_k \mid \mathbf{z}_k)}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1}, x_{1:k})}. \tag{8}$$

Bootstrap filter is applied by using the system model $p(\mathbf{z}_k \mid \mathbf{z}_{k-1})$ as the proposed $p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1}, x_{1:k})$. Then, the posterior distribution $p(\mathbf{z}_{1:k} \mid x_{1:k})$ is recursively obtained as

$$p(\mathbf{z}_{1:k} \mid x_{1:k}) \propto p(\mathbf{z}_{1:k-1} \mid x_{1:k-1}) p(x_k \mid \mathbf{z}_k). \tag{9}$$

For the likelihood $p(x_k \mid \mathbf{z}_k)$, we use

$$p(x_k \mid \mathbf{z}_k) \approx p^{(CC)}(x_k \mid \mathbf{z}_k) \cdot p^{(WCC)}(x_k \mid \mathbf{z}_k) \tag{10}$$

2.3 State Estimation by Particle Filtering

Sequential state estimation is carried out by updating the weighted particles. In each short-term frame, as the first filtering step, normalized sub-band CC functions update the particle weights, and then the particles are

filtered out. The second filtering is done with the weights updated by normalized sub-band WCC functions.

Particles $\{\mathbf{z}_k\}$ with weights $\{w_k\}$ are distributed on spectro-spatial state space, and are updated sequentially according to normalized CC and WCC functions as follows.

$$\{(\mathbf{z}_{k-1}, w_{k-1})\} \xrightarrow{CC} \{(\tilde{\mathbf{z}}_k, \tilde{w}_k)\} \xrightarrow{WCC} \{(\mathbf{z}_k, w_k)\} \tag{11}$$

The two-step particle filtering is performed in the k-th frame as follows:

STEP 1: (particle draw)

Particles are prepared depending on signal energy E_k on bi-dimensional spectro-spatial space and it is defined as:

$$E_k = \sum\nolimits_F \left| X_{k,F} \right|^2$$

if ($k = 1$)

Uniform distribution is adopted.

else if ($E_k \geq$ (threshold) and $E_{k-1} \geq$ (threshold))

Signal has been observed until the previous $(k-1)$-th frame. The particles in the $(k-1)$-th frame, are updated with the weights.

else if ($E_k \geq$ (threshold) and $E_{k-1} <$ (threshold))

Signal onset is detected in this frame. Alternate distribution is prepared based on the histogram of DOA estimates in past. Uniform distribution is adopted in frequency direction. In non-moving multiple sound source conditions, long-term histogram is of great advantage to catch DOAs of new-born sound sources promptly.

else

No process is done, and go to the next $(k+1)$-th frame.

end

STEP 2: (1st step filtering by CC)

The particles are filtered out with the weights given by a set of half-wave rectified sub-band CC functions mainly in frequency direction.

STEP 3: (re-sampling)

Particles $\{\tilde{\mathbf{z}}_k\}$ are re-sampled in proportion to the weight $\{\tilde{w}_k\}$. Distribution of the resampled particles with uniform weights, reflects the energy distribution, and is considered as the proposal particle distribution in the next step.

STEP 4: (2^{nd} step filtering by WCC)

Sub-band GCC functions are half-wave rectified, and are used as the weights for filtering the particles in **STEP 3**. The particles are delivered to the next frame besides the next step.

STEP 5: (re-sampling)

The particles $\{\mathbf{z}_k\}$ with weights $\{w_k\}$ are added along frequency, and global correlation function is obtained throughout the whole frequency.

STEP 6: (finding DOA)

Finally, DOA is estimated from the time lag with the peak in the global correlation function.

3. PERFORMANCE EVALUATION

3.1 Experimental Setup and Conditions

Performance of the proposed method was evaluated by using speech data acquired in a room with computer-generated interferences. Connected digit speech utterances, which were selected from the TI-digit speech database [13], were played through loud speakers (BOSE 101) in a meeting room ($7.0m \times 4.1m \times 2.6m$). In the room, two microphones (audio-technica AT805F) were placed with a spacing of 0.15 m, and two loud speakers were placed at the direction of 0 (the front) or 16 degrees toward the paired-microphone. The received signals were recorded at 48 kHz with 16 bit accuracy using a DAT recorder (SONY TCD-100). The recorded signals were distorted by background noises and room reverberation, since the room was not soundproofed and reverberation time was 0.23 s approximately. Afterwards, independent white Gaussian noises were added to the received speech signals as non-directional distributed measuring noises. All signals were band-limited to 300-3,400 Hz, and the DOA estimation was performed with 3,498 (53 [in lag direction] × 66 [in frequency direction]) particles. DOA estimation was also carried out with both CC and WCC methods as reference.

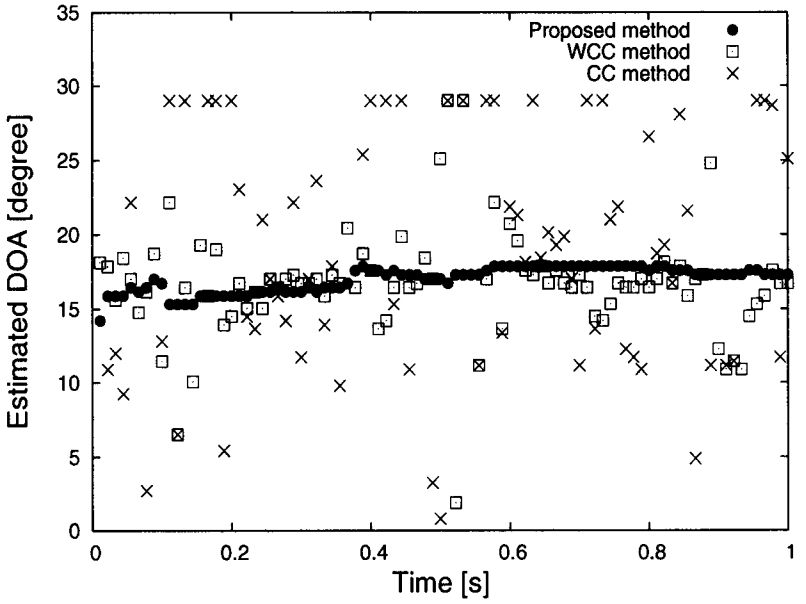


Figure 12-1. DOA estimates: Proposed technique("●"), WCC ("□"), and CC ("×") methods in 10 dB SNR. The true DOA was fixed and set at 16 degrees.

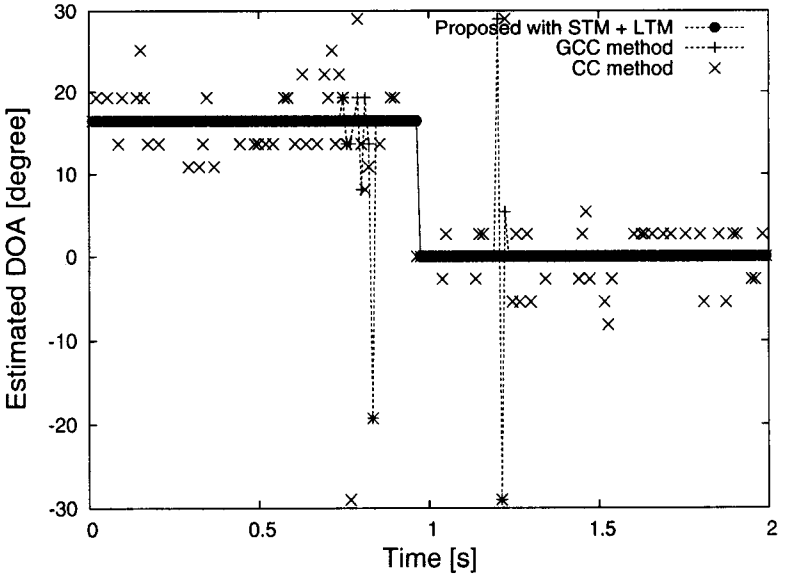


Figure 12-2. DOA estimates in each short-term frame by a grid-based with short-term and long-term memories (STM+LTM) ("●"), WCC ("□"), and CC ("×") methods in 10 dB SNR condition. The proposed method employs proposal distribution given by histogram of the whole DOA estimates in past. The true DOA of the target signal was changed from 16 degrees to 0 degree at 1.0 s.

3.2 Single-Source Condition

A loud speaker was 1.0 m away from the microphones at 16 degrees. Figure 12-1 shows a part of estimated DOAs in each short-term frame by the proposed (marked by "•"), WCC (marked by "□"), and CC (marked by "×") methods in 10 dB SNR. This result indicates that Markov modeling of frame-based DOA estimates helps to make the DOA trajectory smooth without any fatal error. The accurate and smooth DOA trajectories provide considerable benefits for practical applications.

3.3 Multiple-Source Condition

We consider estimating the DOA of the signal from an active source in multiple source condition. Two loud speakers were 1.0 m away from the microphones at 0 and 16 degrees, and the either loud speaker played TI-digit utterances alternately. Firstly, histogram of DOA estimates was formed, and then the DOA histogram is employed as proposal distribution in **STEP 1**.

In vehicles passengers are seated, but they tend to move their heads freely while speaking. Because of this reality, the DOA should be estimated in each short-term frame. Markov model takes the previous DOA estimates into account. In other words, during speech periods, the proposed method uses information stored in a short-term memory.

On the other hands, during non-speech periods, the histogram of previous DOA estimates, that is, a long-term memory, helps to give *a prior* knowledge on rough directions of non-moving speakers. Preparing the histogram of enough DOA estimates, performance of the proposed method is shown in Figure12-2, where grid-based particle filtering is carried out to reduce computational cost. The proposed method achieves the rapid pursuit against sudden DOA changes at 1.0 s.

4. CONCLUSION

This article presents a robust DOA estimation approach a two-step particle filtering in adverse conditions, especially under in-vehicle environments. The proposed method aims at combining the advantages of CC and WCC methods in spectro-spatial domain, which is specified by frequency and time lag. We regard both normalized sub-band CC and WCC functions as likelihoods to obtain the weights for particles on the spectro-spatial space.

The proposed method has a significant advantage over conventional CC and GCC methods in terms of both accuracy and stability. Markov modeling

of the time evolution of frame-based DOA estimates achieves smooth and stable estimation under noisy environments. In multiple source conditions, the proposed method succeeds in tracking the rapid change of DOAs when a long-term histogram of the DOA estimates is adopted as proposal. The in-vehicle environments, where speakers take their seats, are the suitable situation for preparing the DOA histogram.

ACKNOWLEDGMENT

The authors would like to acknowledge Professor Norikazu Ikoma of Kyushu Institute of Technology in Japan and Professor Tomoko Matsui of Institute of Statistical Mathematics also in Japan for their generous suggestions and assistance on particle filtering.

REFERENCES

[1] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, Springer-Verlag, Berlin, 2001.

[2] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, pp. 320 327, 1976.

[3] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001.

[4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," IEEE Transactions on Signal Processing, Vol. 50, Issue 2, pp. 174-188, 2002.

[5] N. Ikoma, N. Ichimura, T. Higuchi, and H. Maeda, "Maneuvering target tracking by using particle filter," Proc. IFSA World Congress and 20th NAFIPS Intl. Conf., Vol. 4, pp. 2223-2228, 2001.

[6] K. V. Tangirala and K. R. Namuduri, "Object Tracking in Video Using Particle Filtering," Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), Vol. 2, pp. 657-660, 2005.

[7] S. Herman and P. Moulin, "A particle filtering approach to FM-band passive radar tracking and automatic target recognition," Proceedings of the IEEE Aerospace Conf., Vol. 4, pp. 1789-1808, 2002.

[8] J. Vermaark and A. Blake, "Nonlinear Filtering for speaker tracking in noisy and reverberant environments," Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), Vol. 5, pp. 3021-3024, 2001.

[9] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill, "Particle methods for Bayesian modeling and enhancement of speech signals," IEEE Transactions on Speech and Audio Processing, Vol. 10, Issue 3, pp. 173-185, 2002.

[10] D. Ward, E. Lehmann, and R. Williamson, "Particle filtering algorithms for tracking an acoustic source in a reverberant environment," IEEE Transactions on Speech and Audio Processing, Vol. 11, pp. 826-836, 2003.

[11] H. Asoh, F. Asano, T. Yoshimura, Y. Motomura, N. Ichimura, I. Hara, J. Ogata, and K. Yamamoto, "An application of a particle filter to Bayesian multiple sound source tracking with audio and video information," Proc. 7th Intl. Conf. on Information Fusion, pp. 805-812, 2004.

[12] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE Trans. Antennas Propagation, Vol. AP-34, pp. 276-280, 1986.

[13] R. G. Leonard, "A database for speaker independent digit recognition," Proceedings of International. Conference on Acoustics, Speech, and Signal Processing (ICASSP '84), Vol. 9, pp. 328-331, 1984.

Chapter 13

NOISE REDUCTION BASED ON MICROPHONE ARRAY AND POST-FILTERING FOR ROBUST SPEECH RECOGNITION IN CAR ENVIRONMENTS

Junfeng Li and Masato Akagi
School of Information Science, Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan

Abstract: Robust speech recognition in a vehicular environment has been an important research field that has attracted great research interest in recent years. Performance of general-purpose speech recognizers dramatically degrade because of various kinds of prevailing noise sources in cars. To deal with acoustic noises, we have proposed two noise reduction systems based on microphone array and post-filtering. In this chapter, we first describe these two noise reduction systems. Then, we present our investigation into the performance improvements of the *automatic speech recognition* (ASR) system for in-car applications when the two noise reduction systems are used as the front-end processors. We report the speech recognition experiments we have conducted using car noise recordings and the AURORA-2J speech database, as well as the recognition results we have obtained. Finally, based on these experimental results, we present some discussions on the proposed noise reduction systems.

Key words: Microphone array; noise reduction; localized noise suppression; non-localized noise suppression; robust speech recognition.

1. INTRODUCTION

Recently there has been increased interest in the hands-free speech processing technology for the in-car environment, such as *automatic speech recognition* (ASR) systems. One main problem associated with this

technology is that the signals received by the distant microphones are severely corrupted by various kinds of noises. Although many algorithms have been published so far [1]-[5], the problem of suppressing noise signals and improving the performance of speech recognition systems in adverse car environments is still a very interesting challenge in the speech signal processing field. A potential solution is to construct a practically effective and computationally efficient noise reduction system as a front-end processor with the goal of improving the performance and robustness of the speech recognition system in car environments.

A variety of noise reduction algorithms for in-car applications have been reported in the literature [1]-[5]. Matassoni *et al.* [2] adopted the single-channel schemes, the magnitude spectral subtraction and the logarithmic *minimum mean square error* (MMSE) estimator, to suppress background noise. The noise-suppressed signals were then used in the speech recognition process, resulting in an improved recognition rate.

Compared to the single-channel technique, the multi-channel technique has shown substantial superiority in reducing noise due to its spatial filtering capability. Zhang *et al.* [4] proposed a "constrained switched adaptive beamformer", which combines a speech adaptive beamformer and a noise adaptive beamformer in a speech/noise constraint selecting scheme, for speech enhancement and recognition in real car environments. However, its relatively slow convergence rate degrades the performance in dealing with non-stationary noise signals in practical conditions. Moreover, Grenier [3] evaluated the performance of the *generalized sidelobe canceller* (GSC) beamformer in car environments. Where he pointed out that the GSC beamformer, as a front-end processor for an ASR system, is not effective in improving the performance of the ASR system in high-noise conditions.

In this chapter, we first study the characteristics of the noise fields in car environments and then introduce two noise reduction algorithms based on microphone array and post-filtering [7]-[12]. The suggested noise reduction algorithms are then evaluated as front-end processors for a speech recognition system to improve the robustness and recognition rate of the system in adverse car environments. Speech recognition results are also reported to show the effectiveness of these two noise reduction algorithms. Some discussions of the two noise reduction systems are finally presented based on the speech recognition results obtained in noisy car conditions.

2. ANALYSIS OF THE NOISE FIELDS IN CAR ENVIRONMENTS

To characterize a noise field, a widely used measure is the *magnitude-squared coherence* (MSC) function, defined as:

$$\Gamma(k,\ell) = \frac{\left|\phi_{x_i x_j}(k,\ell)\right|^2}{\phi_{x_i x_i}(k,\ell)\phi_{x_j x_j}(k,\ell)}, \tag{1}$$

where $\phi_{x_i x_j}(k,\ell)$ is the cross *power spectral density* (PSD) between two signals $x_i(t)$ and $x_j(t)$; $\phi_{x_i x_i}(k,\ell)$ and $\phi_{x_j x_j}(k,\ell)$ are the auto-PSD of $x_i(t)$ and $x_j(t)$, respectively; and k and ℓ are the frequency index and the frame index.

A diffuse noise field has been shown to be a reasonable model for many practical noise environments [6][9][11]. Theoretical MSCs of a perfectly diffuse noise field are plotted in Figure 1, along with the measured MSCs using real-world car noises. From Fig. 1, some characteristics of car noise environments can be easily observed: (1) Car noise environments can be modeled as diffuse noise fields; (2) MSC in car conditions is a frequency-dependent measure; (3) Noises on different microphones are high-correlated in the low frequencies and low-correlated in the high frequencies.

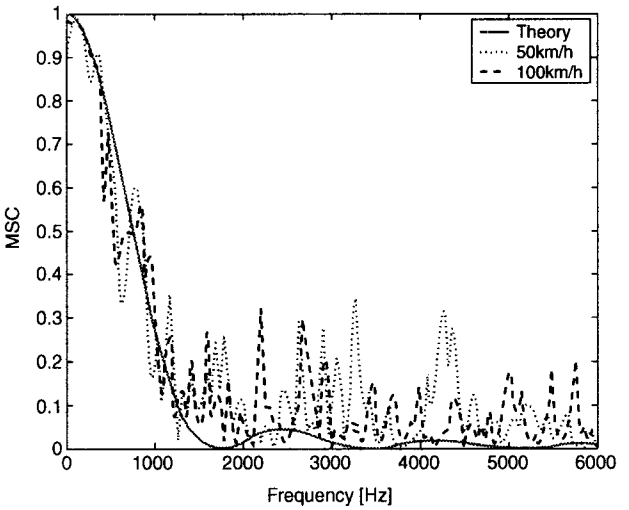


Figure 13-1. Magnitude-squared coherence function in car environments. (The distances between microphones are 10cm.)

3. NOISE REDUCTION ALGORITHMS BASED ON MICROPHONE ARRAY AND POST-FILTERING

Consider an array with three microphones in a noisy environment, as shown in Figure 13-2. The observed signal on each microphone is composed of three components. The first is the desired speech signal from the specific direction of interest. The second is the localized noise, which includes the noise components coming from some determinable directions (point noise sources), e.g., passenger's interference speech and other passing car noise. And the third component is the non-localized noise, which includes the noise components coming from all directions, such as road and engine noise signals in real car environments. The goal of our research was to improve the recognition rate and robustness of a speech recognition system in adverse car environments. The objectives of this research were to reduce both localized and non-localized noises simultaneously, while keeping the desired speech signal distortionless. To implement this idea, we constructed the noise reduction systems, shown in Figure 13-3, which consist of localized noise suppression and non-localized noise suppression.

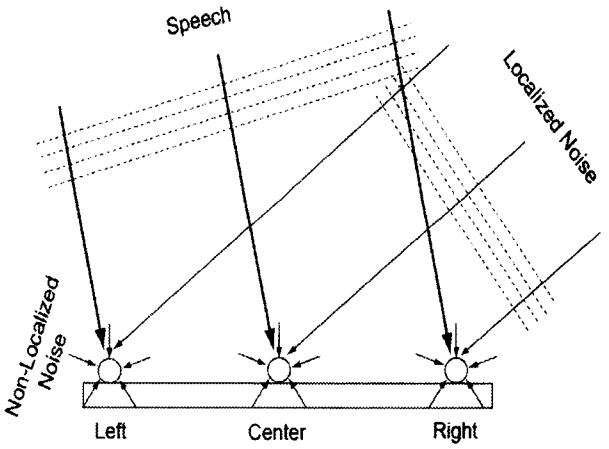


Figure 13-2. Microphone array and signal model.

3.1 Localized Noise Suppression [8][11]

The basic idea of suppressing localized noise is first to estimate the spectra of localized noises and then to subtract the estimated spectra from those of noisy observations. To estimate localized noises, a subtractive beamformer based multi-channel estimation approach was proposed by Akagi et al. [7] that yields highly accurate spectral estimates for localized

noises. The problem of this multi-channel estimation approach is that it fails in some frequencies and some directions that correspond to the grating sidelobes of the small-size microphone arrays.

To solve this problem and improve the performance of this multi-channel estimation approach, the authors have proposed a hybrid noise estimation technique that combines the multi-channel estimation approach and a soft-decision based single-channel estimation approach, producing much more accurate spectral estimates for localized noises [8]. With the use of the hybrid estimation technique, the spectral estimate of localized noise $\hat{N}^c(k,\ell)$ has been calculated by [8],[11]:

$$\hat{N}^c(k,\ell) = \begin{cases} \hat{N}_m^c(k,\ell), & \text{not array nulls} \\ \hat{N}_s^c(k,\ell), & \text{array nulls} \end{cases} \tag{2}$$

where $\hat{N}_m^c(k,\ell)$ and $\hat{N}_s^c(k,\ell)$ are the estimated spectrum for localized noise by the multi-channel estimation approach [7] and the single-channel estimation approach [14], respectively. With this hybrid noise estimation technique, it can be expected that there will be highly accurate spectral estimates for localized noises and that, in a general sense, the grating sidelobes of the microphone arrays with small physical size will be mitigated.

Furthermore, we further enhanced the hybrid noise estimation technique by integrating a *robust and accurate speech absence probability* (RA-SAP) estimator. Considering the strong correlation of speech presence uncertainty between adjacent frequency bins and consecutive frames and making full use of the frequently-perfect high estimation accuracy of the multi-channel approach, we developed the RA-SAP estimator that improves the estimation accuracy of the hybrid noise estimation technique by combining the multi-channel and single-channel approaches in an effective and tight way.

The proposed hybrid noise estimation technique yields highly accurate spectral estimates for localized noises. Subsequently, the spectral estimates are subtracted from those of the observed noisy signals by non-linear spectral subtraction [7][11].

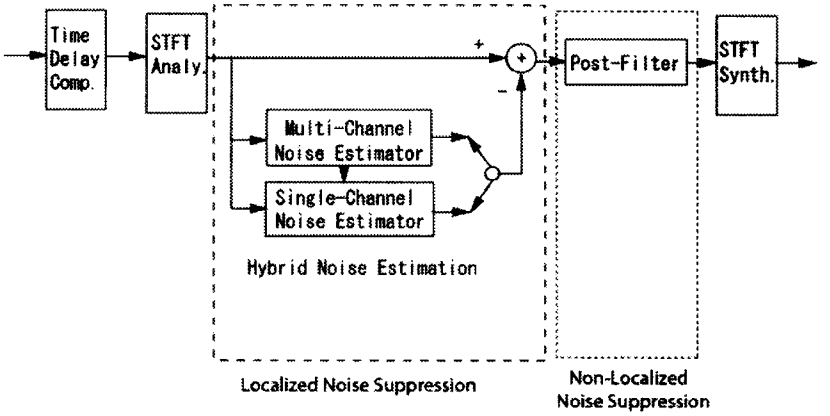


Figure 13-3. Block diagram of proposed noise reduction systems

3.2 Non-localized Noise Suppression

To further suppress non-localized noise, we proposed two post-filters, described as follows.

3.2.1 **Post-filter 1** [8,10]

The first post-filter is based on the *optimally-modified log-spectral amplitude* (OM-LSA) estimator that is characterized by the following gain function [14]:

$$G(k,\ell) = G_{H_1}(k,\ell)^{1-q(k,\ell)} G_{\min}^{q(k,\ell)} \tag{3}$$

where $G_{\min}$, $q(k,\ell)$ and $G_{H_1}(k,\ell)$ are a constraint constant, the *speech absence probability* (SAP) at spectral subtraction output and the gain function of the traditional MMSE-LSA estimator when speech is surely present, as defined in [13].

As Equation 3 shows, the performance of this post-filter is greatly dependent on the SAP, which should be estimated based on the incoming signals. Furthermore, according to Bayes' rule, the success and failure of the SAP estimator are closely related to the *a priori* SAP, which also varies with input signals. To further improve the performance of this post-filter, therefore, we developed a novel estimator for the *a priori* SAP based on the unchanged coherence characteristics of the noise field at spectral subtraction output. Under the assumption of a diffuse noise field, the noise field at spectral subtraction output (i.e., after localized noise suppression) still conforms to the diffuse noise field [9][11].

At spectral subtraction output, therefore, the MSC spectra are subsequently divided into two parts: a high frequency region with low MSCs and a low frequency region with high MSCs, with the transient frequency f_t calculated by $f_t = c/2d$, where c and d are the speed of sound and the distance between microphones, respectively. In the high frequency region, the MSC spectra are further divided into E sub-bands and averaged across the frequencies in each sub-band, after which the average MSC $\overline{\Gamma}_e(k,\ell)$ is obtained in e-th sub-band.

The *a priori* SAP is calculated as follows: if a high averaged coherence (higher than a threshold $T\max_e$) is detected, a speech present state is presumably detected; if a low averaged coherence (lower than a threshold $T\min_e$) is detected, a speech absent state is presumably detected. For the MSCs in $\left[T\min_e,\ T\max_e\right]$, the *a priori* SAPs are determined by the linear interpolation.

In the low frequency region, we calculate an average MSC $\overline{\Gamma}(k,\ell)$, averaged across the frequencies over the transient frequency f_t. Using this average MSC $\overline{\Gamma}(k,\ell)$ and following the ideas similar to the ones used in the high frequency region, we can determine the *a priori* SAPs in the low frequency region.

The estimated *a priori* SAPs are finally incorporated into the post-filter, resulting in the enhanced noise reduction performance of the post-filter [11][12].

3.2.2 **Post-filter 2** [10]

The second post-filter is developed with the assumption of a diffuse noise field to suppress correlated as well as uncorrelated noises. In the proposed post-filter, a modified Zelinski post-filter, which is estimated using the signals on the microphone pairs on which noises are uncorrelated by considering the correlation characteristics of noise impinging on different microphone pairs, is applied to the high frequencies (above the lowest transient frequency) to suppress spatially uncorrelated noise, and a single-channel Wiener post-filter is applied to the low frequencies (below the lowest transient frequency) to cancel of spatially correlated noise [10].

Under the assumption of a diffuse noise field, the spatially weakly correlated noise components only exist in the frequencies over the transient frequency f_t. Since the transient frequencies are only determined by the distances between microphones, microphone pairs with different inter-element spacings are characterized by different transient frequencies. That is, for different microphone pairs with different inter-element spacing, low correlated noise is found in different frequency regions. Furthermore, for a certain frequency, noise is mutually low correlated only on limited

microphone pairs, generally not on all pairs. Therefore, according to the different transient frequencies, we divide the full frequency band into some sub-bands. In each sub-band (except the lowest sub-band), noise signals are mutually weakly correlated for the individual frequency of interest on the microphones of the corresponding pair sets, generally not on all microphone pairs (as used in the Zelinski post-filter). Thus, the spectral densities of desired speech and noisy signal can be estimated from the cross- and auto-PSDs of multi-channel spectral subtraction outputs. Thus, the gain function of the modified Zelinski post-filter is given by:

$$G_{mz}(k,\ell)=\frac{\frac{1}{|\Omega_m|}\sum_{\{i,j\}\in\Omega_m}\Re\left\{\phi_{x_i'x_j'}(k,\ell)\right\}}{\frac{1}{|\Omega_m|}\sum_{\{i,j\}\in\Omega_m}\left[\frac{1}{2}\left(\phi_{x_i'x_i'}(k,\ell)+\phi_{x_j'x_j'}(k,\ell)\right)\right]}, \tag{4}$$

where Ω_m is the corresponding microphone pair set for m-th sub-band, x_i' is the spectral subtraction output in i-th channel.

In the low sub-band, we adopt a single-channel technique to estimate a Wiener filter. The gain function of this Wiener filter is:

$$G_s(k,\ell)=\frac{SNR_{priori}(k,\ell)}{1+SNR_{priori}(k,\ell)}, \tag{5}$$

where $SNR_{priori}(k,\ell)$ is the *a priori* SNR, which is updated in a decision-directed scheme that significantly reduces residual "musical noise" as detailed in [13]. A soft-decision based approach is used to estimate the noise spectrum under speech presence uncertainty [14], which can update the noise estimate even in speech active periods, improving its performance in dealing with non-stationary noise.

4. SPEECH RECOGNITION EXPERIMENTS

The final goal of this investigation was to improve speech recognition performance in car environments. Therefore, we examined the performance of the proposed noise reduction algorithms in terms of recognition accuracy in various car environments.

To evaluate the effects of the proposed noise reduction systems on the performance improvement of a speech recognition system, we have incorporated the systems as front-end processors for a speech recognition

system that performs in adverse car environments. The noisy input signals observed on the multiple microphones were input into the noise reduction systems, yielding the enhanced speech signals. The enhanced speech outputs then were fed into the speech recognition system so that the utterance could be recognized. Thus, the performance improvement of the noise reduction system was evaluated by comparing the system's recognition rate of utterances to the recognition rates obtained by using noisy inputs and other noise reduction algorithms.

4.1 Experimental Configuration

To assess the performance of the studied noise reduction algorithms in terms of speech recognition performance, we performed comprehensive speech recognition experiments.

In the experiments, an equally-spaced linear array consisting of three microphones with inter-element spacing of 10cm was mounted on the roof near the driver's sun-visor in a car. The array was about 50cm away from and directly in front of the driver. Multi-channel noise recordings were performed across all channels while the car was running at the speed of 100km/h with high level air condition noise (the air condition is on).

The training and testing speech signals were selected from the AURORA-2J database [16]. The AURORA-2J is a Japanese version of the AURORA-2, which is a digit strings database.

For testing we generated two sets of noise-corrupted data. The first data set, referred to as *data set A*, involved the addition of randomly selected segments of the multi-channel car noise recordings across 1001 test sentences in the AURORA-2J at different SNR levels from 0dB to 20dB in 5dB steps. The second data set, referred to as *data set B*, involved addition of the multi-channel car noise and a secondary speaker's speech (passenger's interfering noise) that is Japanese digit /ichi/ with DOA of 60 degrees to the right, across 1001 test sentences in AURORA-2J at different SNR levels same as that in *data set A*. Note that *data set B* corresponds to a more realistic context for a typical car environment in which a passenger is speaking. Both *data sets A and B* were used to show the performance of the proposed noise reduction algorithm in car noise conditions. For training an acoustic model, a total of 8400 utterances spoken by 110 speakers (55 male and 55 female speakers) were used.

The signals were pre-emphasized with a coefficient of 0.97. A hamming window of 32ms length with a 16ms frame shift was used. The first 12 dimensions of a de-correlated log compressed Mel energy spectrum were chosen (the zero-th order coefficient was discarded). Combining with the log power energy, we got 13 dimensional static feature vectors. Together with

their first and second order dynamic values, 39 dimensional feature vectors were formed.

The acoustic models consist of ten digits, one silence and short pause models. Each of the digits models has 18 states with 16 output distributions. The silence model has 5 states with 3 output distributions. The short pause model has 3 states with 1 output distribution. Each distribution of the digits models has 20 Gaussians while that of the silence model and the short pause model has 36 Gaussians. Each model was trained as a left-to-right topology with three states (without skip among states) by using Baum-Welch algorithm together with flat-starting embedded training. Standard Viterbi decoding techniques were used for recognition.

4.2 Experimental Results

The noise reduction algorithms we studied, including the delay-and-sum beamformer with Wiener post-filter (DSWF) [15], the hybrid estimation technique based localized noise suppression followed by the post-filter 1 (MA-LSA) [11], and the hybrid estimation technique based localized noise suppression followed by the post-filter 2 (PRO-MAPF) [12], were all assessed using two testing data sets, A and B. The recognition results for two noise reduction algorithms in two noise conditions are presented in Figures 13-4 and 13-5, respectively.

As Figure 13-4 shows, for *data set A*, all tested noise reduction algorithms provide some degree of performance improvement in the speech recognition rate compared with noisy inputs. The DSWF algorithm achieves an average recognition rate improvement of 6.0% with respect to noisy inputs. The MA-LSA provides an average recognition rate improvement of about 13.6%. The PRO-MAPF achieves the highest recognition rate improvement of about 18.6%. The recognition rate improvements drastically increase as the noise level increases (the SNR decreases). Moreover, in very high SNR conditions, all the tested algorithms provided just slight performance improvement compared with the noisy inputs, which is reasonable since the inputs are "clean" enough and a relatively high recognition rate is achievable in these conditions.

Contrasted with the algorithm MA-LSA, the algorithm PRO-MAPF provides a much higher speech recognition rate in all noise conditions. This superiority is the result of the fact that PRO-MAPF introduces low speech distortion, although MA-LSA improved speech quality in subjective evaluations [11].

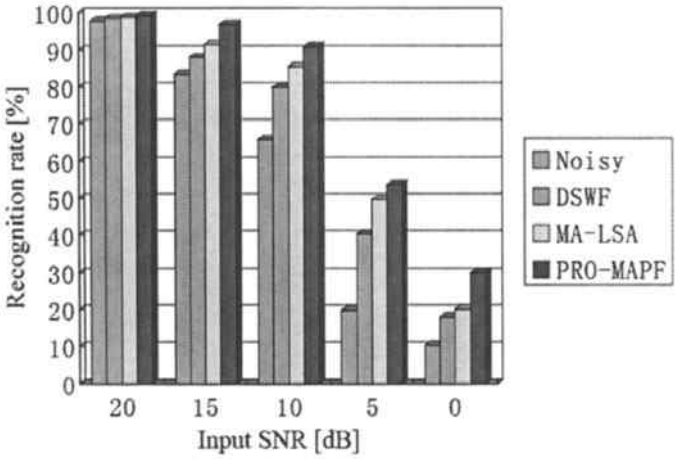


Figure 13-4. Speech recognition results for data set A.

The recognition results for *data set B* are shown in Figure 13-5. We can observe that PRO-MAPF also demonstrates the highest recognition rate at all SNRs. In the noise condition in which the passenger's speech is regarded as localized interfering noise, the recognition rate goes down greatly for unprocessed noisy testing data. Recognition rate improvements of 11.5%, 16.8%, and 23.2% were provided by the DSWF, MA-LSA and PRO-MAPF algorithms, respectively. This highest recognition rate of PRO-MAPF can be attributed to the fact that it is successful in dealing with both the passenger's interfering speech and diffuse car noises simultaneously with minimum speech distortion.

5. DISCUSSIONS

It should be noted that experimental conditions are often slightly different from real-world environments. For example, in the real world, the desired speech signal is corrupted by reverberant noises. In our experiments, reverberation was disregarded. However, in the experimental conditions, the distance between the speaker and the microphone array was about 50cm in the car. In this situation, the speech sound via a direct propagation path is much stronger than those speech sounds via multiple reflected paths. So, the effect of reverberation is very small. Therefore, the results we obtained in experimental conditions are reliable in real-world car environments.

Based on the speech recognition results presented above, it is clear that the PRO-MAPF algorithm using microphone array and post-filter 2 is superior in regard to the other comparable algorithms, DSWF and MA-LSA.

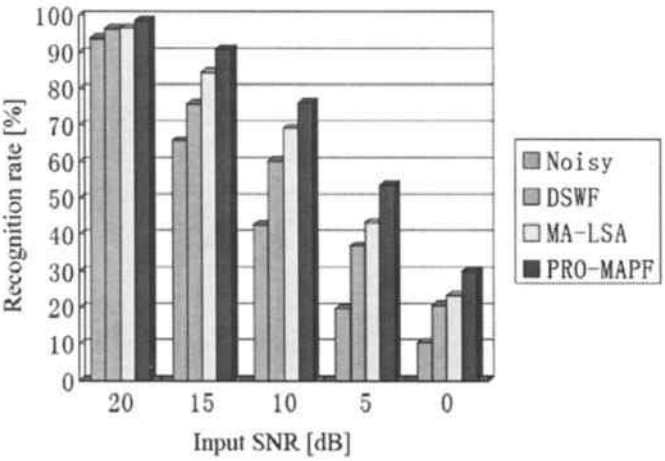


Figure 13-5. Speech recognition results for data set B.

All noise reduction algorithms under consideration have improved recognition accuracy in all tested noise conditions, especially in high noise conditions. In very low noise conditions (high SNR conditions), all signals (input signals and enhanced signals) are relatively "clean" such that the obtained recognition results are relatively high. However, in very high noise conditions (low SNR conditions), the input signals are strongly corrupted by interfering noise signals, resulting in low recognition rates. The signals enhanced by the tested algorithms show relative quality improvement, especially for the PRO-MAPF algorithm, resulting in an improved recognition rate.

Furthermore, the PRO-MAPF algorithm is superior to the DSWF algorithm for other reasons. The DS beamformer can achieve only very limited noise reduction performance for localized noises because of its small physical size (three sensors). The PRO-MAPF algorithm (the hybrid noise estimation technique based noise reduction algorithm) can reduce most localized noise components, theoretically even all localized noises, yielding infinite noise reduction performance in a perfectly coherent noise field. Concerning post-filtering, the traditional Wiener filter used in the DSWF can give only low noise reduction performance because of the involved correlated noise components, especially in the low frequency region. Whereas, in the PRO-MAPF algorithm, the post-filter fully considers and utilizes the spatial correlation characteristics of the noise field, forming a hybrid algorithm which is a combination of a modified Zelinski post-filter in the high frequency region and a single-channel Wiener filter in the low frequency region. This hybrid post-filter suppresses the non-localized noise

components, resulting in the high recognition results shown in Figures 13-4 and 13-5.

Moreover, the PRO-MAPF algorithm also is superior to the MA-LSA algorithm for other reasons. For suppressing localized noise components, both the MA-LSA and the PRO-MAPF algorithms exploited same noise reduction mechanism. In regard to the post-filter for suppressing non-localized noise components, the MA-LSA algorithm considers an OM-LSA estimator based post-filter. Although the OM-LSA based algorithm provides enhanced signals of improved speech quality, some sensitive implementation parameters are not easy to determine, and the non-suitable parameters deteriorate the speech recognition performance in real-world environments. The PRO-MAPF algorithm, on the other hand, avoids the implementation problems and provides a robust solution for implementing the post-filter to suppress non-localized noise components. In short, the PRO-MAPF algorithm reduces non-localized noises very well, which, in turn, improves recognition results in real-world environments, as shown in Figures 4 and 5.

6. CONCLUSION

We have first introduced two noise reduction algorithms that we proposed earlier based on microphone array and post-filtering techniques. We have then showed how the performance of speech recognition systems was improved when the suggested algorithms were used as front-end processors. The speech recognition results using real-world car noise recordings show that the proposed algorithms, MA-LSA and PRO-MAPF, achieve a higher speech recognition rate than the traditional algorithm, DSWF, at all SNRs in all noise conditions; the algorithm PRO-MAPF outperforms the algorithm MA-LSA in improving the recognition rate of an ASR system in all tested environments. This performance improvement can be attributed to the fact that PRO-MAPF is able to deal with various kinds of noise and preserve the speech components (low speech distortion) simultaneously. Whereas, the algorithm MA-LSA is suitable for reducing noise in speech communication systems due to the high quality of enhanced signals which was confirmed by the listening tests in [11].

ACKNOWLEDGEMENT

The authors would like to thank Dr. Xugang Lu for his kind help in performing speech recognition experiments and acknowledge Professor

Joerg Bitzer of University of Applied Sciences in Oldenburg, Germany, for his constructive discussions throughout this work.

REFERENCES

[1] A. Mrutti, P. Coteetti, et al., "On the development on an in-car speech interaction system at IRST", In Proc. of Special Workshop in Maui (SWIM), 2004.

[2] M. Matassoni, M. Omologo and C. Zieger, "Experiments of in-car audio compensation for hands-free speech recognition", In Proc. IEEE Workshop on Automatic speech recognition and understanding, 2003.

[3] Y. Grenier, "A microphone array for car environments", Speech Communication, vol. 12, no. 1, pp. 25-39, 1993.

[4] X.X. Zhang and John H.L. Hansen, "CSA-BF: A constrained switched adaptive beamformer for speech enhancement and recognition in real car environments", IEEE Trans. On speech and audio processing, vol. 11, no 6, pp.733-744, 2003.

[5] M. Nakayama, et al., "An evaluation of in-car speech enhancement techniques with microphone array steering", In Proc. ICA2004, pp. 3041-3044, 2004.

[6] M. Brandstein and D. Ward (eds.), *Microphone Arrays: Signal Processing Techniques and Applications*, Springer-Verlag, Berlin, 2001.

[7] M. Akagi and T, Kago. "Noise reduction using a small-scale microphone array in multi noise source environment", In Proc. ICASSP2002, pp. I909-912, 2002.

[8] J. Li and M. Akagi, "Noise reduction using hybrid noise estimation techniques and post-filtering", In Proc. ICSLP2004, pp. 2705-2708, 2004.

[9] J. Li, X. Lu and M. Akagi, "A noise reduction system in arbitrary noise environments and its applications to speech enhancement and speech recognition", In Proc. ICASSP2005, pp. 277-280, 2005.

[10] J. Li and M. Akagi, "A hybrid microphone array post-filter in diffuse noise field", In Proc. Eurospeech2005, pp. 2313-2316, 2005.

[11] J. Li and M. Akagi, "A noise reduction system based on hybrid noise estimation technique and post-filtering in arbitrary noise environments", To appear in Speech Communication, 2005.

[12] J. Li, Noise reduction based on microphone arrays and post-filtering for robust hands-free speech recognition in adverse environment, Unpublished Ph.D thesis, Japan Advanced Institute of Science and Technology, Japan, March, 2006.

[13] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator", IEEE Trans. on Acoustic. Speech and Signal Processing, vol. 33, no. 2, pp.443-445, 1985.

[14] I. Cohen and B. Berduo, "Speech enhancement for non-stationary noise environments", Signal processing, vol. 81, no. 11, pp. 2403-2418, 2001.

15] K. U. Simmer and A. Wasiljeff, Adaptive microphone arrays for noise suppression in the frequency domain. In Proc. Workshop on Adaptive Algorithms in Communications, pp. 185-94, 1992.

[16] http://sp.shinshu-u.ac.jp/CENSREC

Chapter 14

ICA-BASED TECHNIQUE IN AIR AND BONE-CONDUCTIVE MICROPHONES FOR SPEECH ENHANCEMENT

Zhipeng Zhang, Kei Kikuiri, Nobuhiko Naka, and Tomoyuki Ohya
Multimedia Laboratories, NTT DoCoMo 3-5 Hikari-no-oka, Yokosuka, Kanagawa, 239-8536 Japan. Email: zzp@mml.yrp.nttdocomo.co.jp

Abstract: How to obtain clean speech signal in noisy environments is a crucial issue for improving the performance of mobile phones. We propose to supplement the existing normal air-conductive microphone with a bone-conductive microphone for noise reduction. We propose to apply the ICA (Independent Component Analysis)-based technique to the air and bone-conductive microphone combination for speech enhancement. The speech signal output by the bone-conductive microphone has the advantage of very high SNR, which well supports the generation of a clean speech signal in combination with a normal microphone. We evaluate this method by a Japanese digital recognition system. The results confirm that the proposed method can allow a mobile phone to obtain a clean speech signal even if the background noise is relatively high.

Key words: ICA; bone-conductive speech; speech enhancement

1. INTRODUCTION

Speech capturing devices are usually disturbed by diffused noise, especially in mobile phone applications where devices are frequently used in noisy environments. In highly adverse conditions, the ambient noise needs to be combated. How to obtain a clean speech signal in noisy environments is a crucial issue if mobile phone communication is to be more widely adopted. During the last decade many research studies have been carried out in this area. Enhancement based on Spectral Subtraction (SS) is the most frequently employed technique[1]. Albeit its simplicity and wide-spread usage, the main disadvantage of SS is that it fails to handle time varying noise well. To

achieve even better performance, a technique called SPIRIT offers the two-microphone NC (noise canceling) solution [2]: one microphone close to and the other far from the speaker. The first one picks up both the desired signal and the background noise, while the other microphone picks up just the noise. By subtracting the second signal from the first, the speech signal is cleaned due to cancellation of the noise signal. In comparison with the single-microphone SS method, SPIRIT provides better sound quality and suppresses the annoying noise. However, it is difficult to catch a pure noise-only signal given that the user's voice can vary so widely; this problem restricts the degree of quality enhancement possible. Other two-microphone noise canceling techniques have been proposed[3,4]. They estimate a weighting factor for the two-channel signal using the optimization of an appropriate energy criterion[3] or diversity method[4] and better performance has been reported. However, these methods fail to deal with non-stationary noise.

In order to obtain clean speech under non-stationary noisy conditions, good performance has been reported with beamforming or blind source separation (BSS) such as Independent Component Analysis (ICA)[5-7]. ICA seems attractive and promising but it is difficult to achieve the desired speech signal in highly adverse conditions [8].

This paper proposes using a bone-conductive microphone to supplement the existing normal air-conductive microphone of a mobile phone for noise reduction. The bone-conductive microphone outputs a speech signal that has higher SNR than that provided by normal microphones. Speech enhancement by combining air and bone-conductive microphones appears to be a very promising approach.

Liu et al. proposed a speech enhancement method using air and bone-conductive microphones[9], where they used the direct filtering method for standard and throat microphones in a noisy environment, and very good results were reported. There are many differences between their work and the proposed technique in this chapter. In their paper[9], they assumed that the noise signals in the air microphone and bone channels were zero-mean Gaussian random variables. To estimate these parameters, non-speech frames need to be detected by a speech/non-speech detector module. System performance largely depends on the accuracy of voice activity detection (VAD) and it fails when the number of detected non-speech frames is insufficient. Another problem is that the noise in real environments is not always a zero-mean Gaussian random variable; a speech enhancement method that can cope with more general environments is needed.

In this chapter, we propose an ICA-based technique to create an effective air and bone-conductive microphone arrangement for speech enhancement. First we explain the signals acquired through the air and bone-conductive microphones in mobile communication systems, and then propose the ICA

technique for processing noisy speech signals. In the next section, we report some experiments and evaluation by speech recognition. The paper concludes with a general discussion and issues related to future research.

Figure 14-1. Microphone arrangement in mobile device.

2. SIGNAL ACQUISITION BY AIR AND BONE-CONDUCTIVE MICROPHONES IN MOBILE DEVICE

2.1 Air and Bone-conductive Microphones for Mobile Phone

In mobile phone use, the ideal placement of air and bone-conductive microphones is shown in Figure 14.1. The bone-conductive microphone is fixed near the top of mobile phone. When speaking, the bone-conductive signal can be recorded when the mobile phone, held by the subject's fingers, is pressed to his face (Figure 14-2); the normal air microphone picks up the speech signal near mouth. This use style strongly supports integration with mobile terminals.

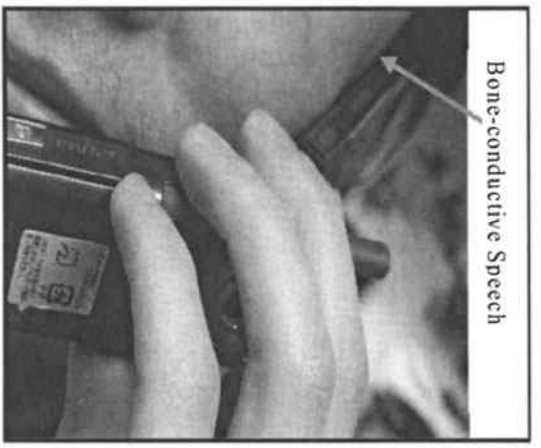


Figure 14-2. Signal acquisition by air and bone-conductive microphones.

2.2 Properties of Speech Signals Captured by Air and Bone-conductive Microphones

Compared to the air-conductive microphone, the bone-conductive microphone is insensitive to ambient noise, but the high frequency portion of the speech signal is insufficient. Therefore we can not directly use bone-conductive microphone for mobile communication even though they have high SNR; this is in contrast to the air-conductive microphone which yields a full frequency signal with low SNR. By combining the two speech signals, we are able to significantly suppress the background noise and obtain a signal that covers the full frequency range and has high SNR.

3. ICA-BASED TECHNIQUE

3.1 Concept of ICA

ICA is a statistical method that decomposes multivariate data into a linear sum of non-orthogonal basis vectors. ICA is widely used for multi-channel signal processing[5-7]. Assuming that the original signals are independent we can apply an ICA algorithm to blindly recover the unknown sources. Here we consider a 2*2 mixing matrix; the source signal $S(\omega)$ is altered by the mixing matrix $A(\omega)$ in the frequency domain. We thus have

$$Y(\omega) = A(\omega)S(\omega). \quad (1)$$

where ω denotes frequency and $Y(\omega)$ denotes the observation signal. Here $A(\omega)$ corresponds to the frequency response of the mixing filters, from the

source to the microphone. The source signal can be recovered if we know the frequency responses of all sources to all microphones and we can calculate the inverse of A(ω). However, in most cases we have no information about the source or indeed the microphone. The problem in recovering original signal S, given only sensor outputs Y(ω), is to obtain Z(ω) by estimating the un-mixing matrix W blindly:

$$Z(\omega) = W(\omega)Y(\omega). \tag{2}$$

The basic assumption of ICA is that the source components are, at each time instant, mutually independent and that each component is white, i.e.: there are no dependencies between time points. This assumption usually holds for speech signals in the real world[10]. ICA has bees successfully used in microphone arrays for speech enhancement. However, in ICA-based multi-channel speech signal processing, it is difficult to extract the desired source signal given the highly adverse conditions. We must focus on developing a robust method that is specific to the application intended. By applying ICA to a bone-conductive speech signal, which has higher SNR, it is easy to extract the desired speech signal.

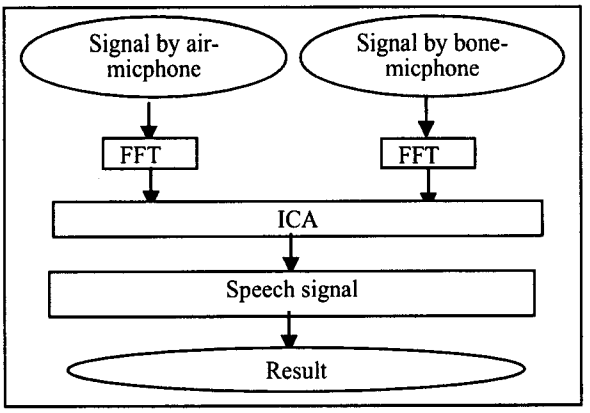


Figure 14-3. System flow chart.

3.2 Learning Rule

W is learnt by maximizing the likelihood (ML) so that

$$\hat{W} = \arg\max_{w} \sum_{t=1}^{T} \log\ p\{y(t); w\} \tag{3}$$

The general learning rule is:

$$W_{i+1} = W_i + \eta[I - \frac{d\log p(y)}{dy} y^T]W_i \quad (4)$$

where I is the identity matrix and $p(y)$ is the probability function of y. This learning rule uses the natural gradient extension as described in[11]. The problem with ML-based estimation is that the convergence speed is slow and it is difficult to extract the desired signal in highly adverse conditions.

A problem with frequency-domain processing is permutation. We need to align the permutation in each frequency bin so that the separated signal in the time domain contains frequency components from the same source signal. We adopted the method introduced by Asano[12] to solve the permutation problem. After ICA was performed, enhanced speech signal is obtained as well as the noise signal. Figure 14-3 depicts the system flow.

4. EXPERIMENTS

4.1 Task and Data

We have performed experiments to evaluate the effectiveness of our proposed method on speech recognition. The task of the system was the recognition of connected Japanese digits, each having 2-8 digits, such as "3429" and "246858". In a preliminary experiment, we have used a bone-conductive throat microphone instead of the one placed near the top of the cell phone device. Speech uttered by a single speaker was simultaneously captured by air and bone-conductive microphones simultaneously. 30 utterances of this speaker were recorded at 16 kHz with 16-bit resolution in a clean condition (room). The speech signal was corrupted by adding noisy data ("Exhibition hall"). The noisy data were simulated at three SNR conditions: 0,5,10 dB. The noise data were non-stationary.

4.2 Feature Vector and HMM

The speech signals were converted into a 25-dimensional acoustic vector consisting of 12-dimensional cepstral-mean-normalized MFCCs and their first derivatives, as well as normalized log energy coefficients. The HMM used in our experiments was a 5 state left-to-right HMM, each state has 4 mixtures.

4.3 Spectrogram of Extracted Speech Data by ICA

ICA was performed to extract clean speech. The length of FFT is 512. The number of iteration in ICA is 1 as to reduce the computational time. Figures 14-4 and 14-5 depict samples of the noisy signals captured by the air and bone-conductive microphones. We can observe that the high frequency portion of the bone-conductive speech is not ample. Figure 14-6 shows the signal by ICA. It is clear that this signal not only has a higher SNR but also improves ampleness at high frequency portions.

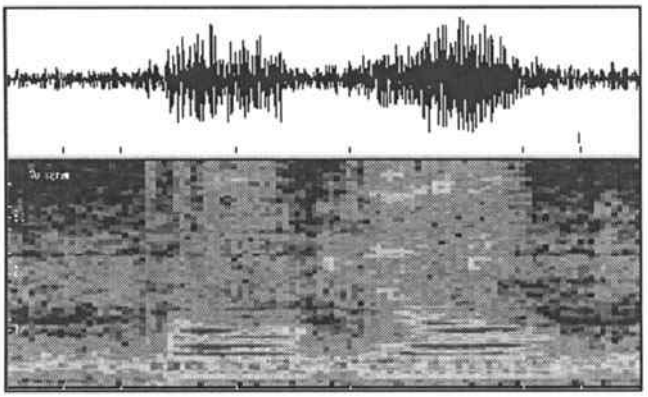


Figure 14-4. Spectrogram of speech data by air-conductive microphone.

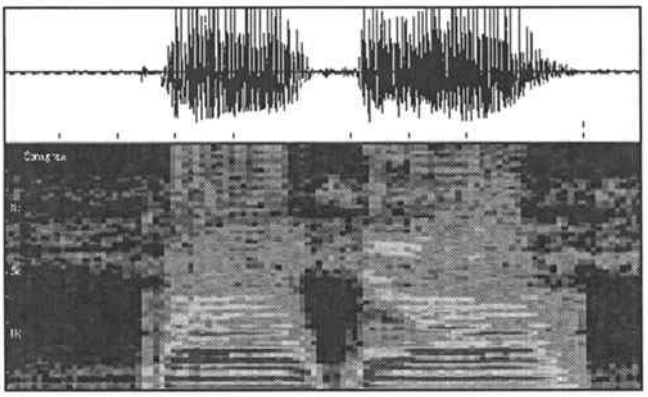


Figure 14-5. Spectrogram of speech data by bone-conductive microphone.

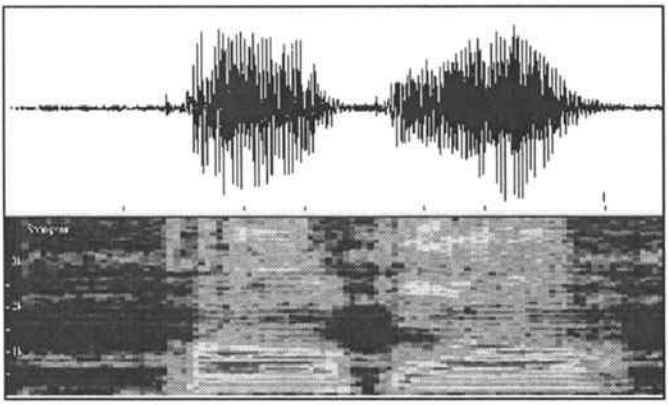


Figure 14-6. Spectrogram of speech data by ICA.

Table 14-1. Comparison of word accuracy in percent achieved with air microphone only, bone-conductive microphone only, and ICA.

	0dB	5dB	10dB
Air-microphone	39.6	58.4	82.6
Bone-microphone	43.5	62.7	85.4
ICA	49.3	66.7	88.1

4.4 Recognition Results

Recognition experiments were performed and Table 14-1 shows the comparison result (accuracy %) for the three arrangements: air microphone only, bone-conductive microphone only, and ICA output. The proposed method achieved a 4.8% and 7.8% improvement in word accuracy compared to air and bone-conductive microphone arrangements. This demonstrates the effectiveness of the proposed method.

5. CONCLUSION

In this chapter, we have presented a new method of using a bone-conductive microphone to supplement an existing normal air-conductive microphone for noise reduction in mobile communication devices. We proposed a ICA-based technique to process the outputs of the air and bone-conductive microphone combination and so enhance speech quality. The proposed method has several advantages: it does not need voice activity

detection (VAD), it can handle the noise of real environments, and the physical arrangement of the microphones is practical.

By observing the spectrogram of speech signal obtained by ICA, we found that the proposed method improves speech quality. It not only has a higher SNR but also is ample at high frequency portions. The proposed method has also been evaluated by a speech recognition test. Recognition results show that the proposed method improves the recognition accuracy compared to the air microphone only and the bone-conductive microphone only arrangements, respectively. Future research includes increasing the variation of test data, investigating the effects of speaker variation, and adding a compensation technique that improves the quality of bone-conductive speech.

ACKNOWLEDGEMENT

This research has been conducted in cooperation with Furui Laboratory at Tokyo Institute of Technology, Japan. In particular, the authors wish to express their thanks to Professor Sadaoki Furui and Dr. Koji Iwano for their assistance and valuable discussions.

REFERENCES

[1] S. Boll, Suppression of Acoustic Noise in Speech Using Spectral Subtraction, IEEE Transactions on ASSP, vol. 27, No. 2, pp. 113-120, (1979).

[2] A. Guerin, R. Le Bouquin, and G. Faucon, A Two-Sensor Noise Reduction System: Applications for Hands-free Car Kit, in Proc. EURASIP JASP, pp.1125-1134, (2003).

[3] S. Srinivasan, M. Nilsson and W. B. Kleijn, Denoising Through Source Separation and Minimum Tracking, in Proc. EUROSPEECH, pp.2349-2352, (2005).

[4] J.Freudenberger and K. Linhard, A Two-Microphone Diversity System and its Application for Hands-Free Car Kits, in Proc. EUROSPEECH, pp.2249-2332, (2005).

[5] K.Torkkola,. Blind Separation of Convolved Sources based on Information Maximization. In Workshop on Neural Networks for Signal Processing, pp.423–432, (1996).

[6] F. Ehlers, and H.Schuster, Blind Separation of Convolutive Mixtures and an Application in Automatic Speech Recognition in Noisy Environment. IEEE Transactions on Signal processing, 45(10): pp.2608–2609. (1997).

[7] T.-W.Lee, A.Bell, and R. Lambert, Blind Separation of Convolved and Delayed Sources. In *Advances in Neural Information Processing Systems*, pp.758–764. MIT Press. (1997).

[8] D. Saitoh, A. Kaminuma, H. Saruwatari, T.Nishikawa and A. Lee, Speech Extraction in a Car Interior using Frequency-Domain ICA with Rapid Filter Adaptations, in Proc. EUROSPEECH, pp.2301-2304, (2005).

[9] Z. Liu, Z Zhang, A. Acero, J. Droppo, X. Huang, Direct Filtering for Air- and Bone-Conductive Microphones, in Proc. IEEE MMSP, (2004).

[10] S. Makino, S. Araki, R. Mukai, and H. Sawada, ICA-based Audio Source Separation, in Proc. International Workshop on Microphone Array Systems - Theory and Practice, (2003).

[11] A. Bell, and T.Sejnowski,. An Information Maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation, pp.1129–1159, (1995).

[12] F. Asano, S. Ikeda, M. Ogawa, H. Asoh, and N. Kitawaki, "A combined approach of array processing and independent component analysis for blind separation of acoustic signals," in Proc. ICASSP, pp.2729–2732, (2001).

Chapter 15

ACOUSTIC ECHO REDUCTION IN A TWO-CHANNEL SPEECH REINFORCEMENT SYSTEM FOR VEHICLES

Alfonso Ortega, Eduardo Lleida, Enrique Masgrau, Luis Buera, and Antonio Miguel

Communication Technologies Group (GTC). Aragón Institute for Engineering Research (I3A). University of Zaragoza, Spain.

Abstract: This chapter presents a two-channel speech reinforcement system which has the goal of improving speech intelligibility inside cars. As microphones pick up not only the voice of the speaker but also the reinforced speech coming from the loudspeakers, feedback paths appear. These feedback paths can make the system become unstable and acoustic echo cancellation is needed in order to avoid it. In a two-channel system, two system identifications must be performed for each channel, one of them is an open-loop identification and the other one is closed-loop. Several methods have been proposed for echo suppression in open-loop systems like hands-free systems. We propose here the use of echo suppression filters specially designed for closed-loop subsystems along with echo suppression filters for open-loop subsystems based on the optimal filtering theory. The derivation of the optimal echo suppression filter needed in the closed-loop subsystem is also presented.

Key words: Speech Reinforcement; echo cancellation; acoustic feedback reduction.

1. INTRODUCTION

A speech reinforcement system can be used in medium and large size motor vehicles to improve the communications among passengers (Ortega et al., 2005; Gallego et al., 2002). Inside a car, speech intelligibility can be degraded due to the lack of visual contact between speaker and listener, the noise and the use of sound absorbing materials among other factors. Using a set of microphones placed on the ceiling of the car, this system picks up the

speech of each passenger. After that, it is amplified and played back into the cabin using the loudspeakers of the audio system of the car.

Acoustic echo appears because the signal radiated by the loudspeakers is picked up again by the microphones. Due to the amplification stage between the microphones and the loudspeakers, the system can become unstable.

Along with the speech signal, the noise is also picked up by the microphones and amplified by the system increasing the overall noise level present inside the car. To prevent this, a noise reduction stage must be used.

Echo Cancellers (AEC) are widely used to overcome electro-acoustic coupling between loudspeakers and microphones (Breining et al., 1999). In a two-channel system, each channel must have two echo cancellers, one corresponding to an open-loop subsystem and the other one corresponding to a closed-loop subsystem. Nevertheless, to achieve enough echo attenuation the use of Echo Suppression Filters (ESF) is needed. Several techniques have been proposed for further echo attenuation using residual echo reduction filters (Gustafsson et al., 1998; Hänsler and Schmidt, 2000). These techniques can be used for open-loop systems but in a speech reinforcement system for vehicles, the ESF must also ensure stability in the closed-loop subsystems. The study for a one-channel system can be found in (Ortega et al., 2003) and the optimal ESF transfer function for the closed-loop subsystems in a two-channel speech reinforcement system is derived here.

Another important aspect of this system is that the overall delay must be short enough to achieve full integration of the sound coming from the direct path and the reinforced speech coming from the loudspeakers.

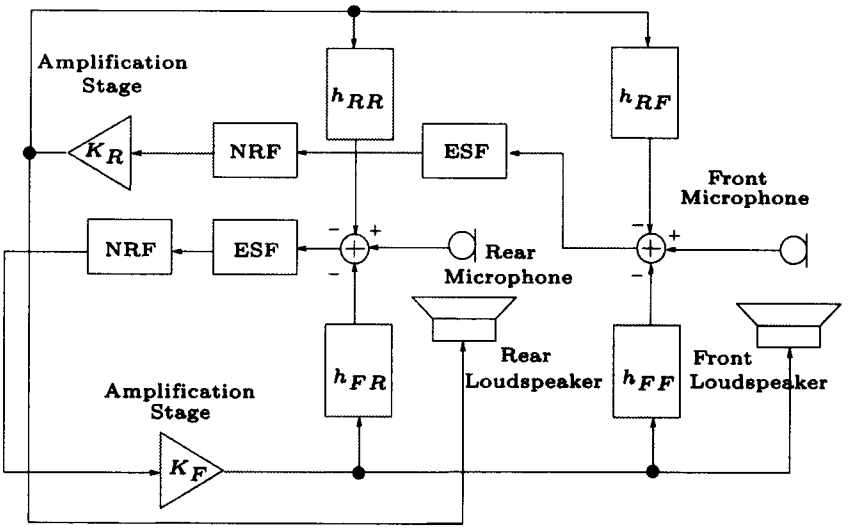


Figure 15-1. Schematic diagram of a two-channel speech reinforcement system for cars.

2. DESCRIPTION AND STABILITY STUDY OF THE TWO-CHANNEL SYSTEM

In order to make communications inside a car more comfortable, a two-channel speech reinforcement system is required. One channel must take the speech of the rear passengers to the front part of the car and the other one must take the speech of the front passengers to the rear seats. A block diagram of the two-channel system is presented in Figure 15-1.

In a two-channel speech reinforcement system, for each channel, there must be two echo cancellers, an echo suppression filter, a Noise Reduction Filter (NRF) and an amplification stage.

The estimation of each Loudspeaker-Enclosure-Microphone (LEM) path performed by each adaptive filter is not enough to ensure the stability of the system. The inaccuracy of the estimation can make the system become unstable. The transfer function of the channel from microphone X (rear, R, or front, F) to loudspeaker Y (F or R) is

$$P_{XY}\left(e^{j\omega}\right)=\frac{K_Y W_Y\left(e^{j\omega}\right)\left[1-K_X W_X\left(e^{j\omega}\right)\tilde{H}_{XY}\left(e^{j\omega}\right)\right]}{D\left(e^{j\omega}\right)} \tag{1}$$

where

$$\begin{aligned} D\left(e^{j\omega}\right)=1&-K_F W_F\left(e^{j\omega}\right)\tilde{H}_{FR}\left(e^{j\omega}\right)-K_R W_R\left(e^{j\omega}\right)\tilde{H}_{RF}\left(e^{j\omega}\right)\\ &-K_R K_F W_R\left(e^{j\omega}\right)W_F\left(e^{j\omega}\right)\tilde{H}_{FF}\left(e^{j\omega}\right)\tilde{H}_{RR}\left(e^{j\omega}\right)\\ &+K_R K_F W_R\left(e^{j\omega}\right)W_F\left(e^{j\omega}\right)\tilde{H}_{RF}\left(e^{j\omega}\right)\tilde{H}_{FR}\left(e^{j\omega}\right) \end{aligned} \tag{2}$$

and $\tilde{H}_{XY}(e^{j\omega})$ is the difference between the LEM path transfer function $H_{FR}(e^{j\omega})$ and its corresponding adaptive filter transfer function $\hat{H}_{XY}(e^{j\omega})$. $W_R(e^{j\omega})$ is the transfer function of the system composed of the ESF and the NRF for the front-rear channel and $W_F(e^{j\omega})$ for the rear-front channel. K_F and K_R are the gain factors for the rear-front channel and the front-rear respectively.

The optimal transfer function each channel is

$$P_{XY}\left(e^{j\omega}\right)=K_Y W_{Yn}\left(e^{j\omega}\right) \tag{3}$$

where $W_{Yn}(e^{j\omega})$ is the transfer function of the noise reduction filter of the corresponding channel. Substituting (3) into (1), and considering (2), the optimal echo suppression filter follows

$$W_{Xe}\left(e^{j\omega}\right)=\frac{W_{Ye}\left(e^{j\omega}\right)}{D_{Xe}\left(e^{j\omega}\right)} \tag{4}$$

with

$$\begin{aligned} D_{Xe}\left(e^{j\omega}\right)=1-K_Y W_{Yn}\left(e^{j\omega}\right)W_{Ye}\left(e^{j\omega}\right)\tilde{H}_{YX}\left(e^{j\omega}\right) \\ +K_X W_{Xn}\left(e^{j\omega}\right)W_{Ye}\left(e^{j\omega}\right)\tilde{H}_{XY}\left(e^{j\omega}\right) \end{aligned} \tag{5}$$

The optimal expressions for both echo suppression filters are not independent and must be fulfilled simultaneously to ensure unconditional stability. This is only possible if

$$K_R\tilde{H}_{RF}\left(e^{j\omega}\right)=K_F\tilde{H}_{FR}\left(e^{j\omega}\right), \tag{6}$$

for each frequency, which implies that both filters must be equal to each other

$$W_{Re}\left(e^{j\omega}\right)=W_{Fe}\left(e^{j\omega}\right). \tag{7}$$

The condition in (6) is not under the control of the designer, so it could not be always met.

3. ECHO SUPPRESSION FILTERS FOR THE CLOSED-LOOP SUBSYSTEMS AND THE OPEN-LOOP SUBSYSTEMS

One possible solution to increase the stability of the two-channel speech reinforcement system is to distinguish between open-loop subsystems and closed-loop subsystems applying specific treatment approaches to each one of them.

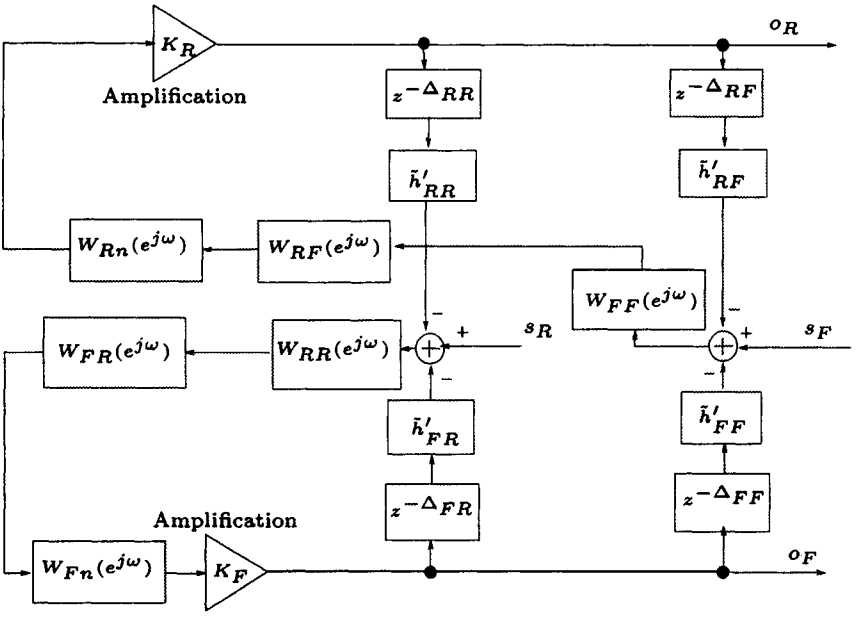


Figure 15-2. Two-channel speech reinforcement system with differentiated treatment techniques for closed-loop subsystems and for open-loop subsystems.

To cope with the residual echo remaining after the echo canceller for the open-loop subsystems, several approaches have been proposed in the literature (Gustafsson et al., 1998; Hänsler and Schmidt, 2000; Enzner et al., 2002). The use of the filters $W_{FF}(e^{j\omega})$ and $W_{RR}(e^{j\omega})$, that follow a Wiener based approach, is proposed.

In order to increase the stability margin of the speech reinforcement system, we propose here the use of the echo suppression filters $W_{RF}(e^{j\omega})$ and $W_{FR}(e^{j\omega})$, specially designed for the closed-loop subsystems.

The proposed system is presented in Figure 15-2. where s_R and s_F are the input signals for the rear-front channel and the front-rear channel respectively, o_R is the output signal of the front-rear channel and o_F is the output signal of the rear-front channel. Due to the propagation delay, the LEM path of each loudspeaker-microphone pair is modelled as a delay block of Δ_{XY} samples followed by a linear system with the same impulse response of the LEM path except for the first Δ_{XY} values. The first Δ_{XY} coefficients of its corresponding adaptive filter are also set to zero to compensate for the propagation delay.

According to Figure 15-2, the transfer functions of the system from microphone X to loudspeaker Y follow:

$$P_{XY}\left(e^{j\omega}\right)=\frac{L_X\left(e^{j\omega}\right)\left[1-L_Y\left(e^{j\omega}\right)W_{XXe}\left(e^{j\omega}\right)\tilde{H}_{XY}\left(e^{j\omega}\right)\right]}{D_2\left(e^{j\omega}\right)} \tag{8}$$

with

$$L_X\left(e^{j\omega}\right)=K_X W_{XYe}\left(e^{j\omega}\right)W_{Xn}\left(e^{j\omega}\right)W_{YYe}\left(e^{j\omega}\right) \tag{9}$$

$$\begin{aligned}D_2\left(e^{j\omega}\right)=1&-L_F\left(e^{j\omega}\right)\tilde{H}_{FR}\left(e^{j\omega}\right)-L_R\left(e^{j\omega}\right)\tilde{H}_{RF}\left(e^{j\omega}\right)\\&-L_F\left(e^{j\omega}\right)L_R\left(e^{j\omega}\right)\tilde{H}_{RR}\left(e^{j\omega}\right)\tilde{H}_{FF}\left(e^{j\omega}\right)\\&+L_F\left(e^{j\omega}\right)L_R\left(e^{j\omega}\right)\tilde{H}_{RF}\left(e^{j\omega}\right)\tilde{H}_{FR}\left(e^{j\omega}\right)\end{aligned} \tag{10}$$

where

$$W_{XYe}\left(e^{j\omega}\right)=\frac{1}{1+K_X W_{YYe}\left(e^{j\omega}\right)W_{Xn}\left(e^{j\omega}\right)\tilde{H}_{XY}\left(e^{j\omega}\right)} \tag{11}$$

is the transfer function of the proposed ESF for the closed-loop subsystems. Substituting (11) into (8), the system transfer functions satisfy

$$P_{XY}\left(e^{j\omega}\right)=\frac{K_X W_{Xn}\left(e^{j\omega}\right)W_{XXe}\left(e^{j\omega}\right)}{D_3\left(e^{j\omega}\right)} \tag{12}$$

where

$$\begin{aligned}D_3\left(e^{j\omega}\right)=1&-K_R W_{RRe}\left(e^{j\omega}\right)W_{Rn}\left(e^{j\omega}\right)\tilde{H}_{FF}\left(e^{j\omega}\right)\\&\times K_F W_{FFe}\left(e^{j\omega}\right)W_{Fn}\left(e^{j\omega}\right)\tilde{H}_{RR}\left(e^{j\omega}\right)\end{aligned} \tag{13}$$

Thus, the stability of the reinforcement system, assuming that the echo suppression filters are working properly, depends only on the open-loop subsystems. That is, the stability depends on the misadjustment functions $\tilde{H}_{RR}(e^{j\omega})$ and $\tilde{H}_{FF}(e^{j\omega})$ that is intended to be minimized by the filters $W_{RRe}(e^{j\omega})$ and $W_{FFe}(e^{j\omega})$ respectively.

The echo suppression filters for the closed-loop subsystems, that increase the stability of the two-channel reinforcement system, depend on the

misadjustment functions of the closed-loop subsystems that are a priori unknown. Assuming that the ESF for the open-loop subsystems are real valued functions, as well as the NRF for each channel, it can be shown that using the magnitude of the misadjustment function is the best option to increase the stability of the system[1].

The estimates of the magnitude of the misadjustment function for each closed-loop subsystem are obtained using estimates of the residual echo $r_{FF}(n)$ for the rear-front channel and estimates of the residual echo $r_{RR}(n)$ for the front-rear channel.

For the front-rear channel, the residual echo remaining after the closed-loop subsystem acoustic echo canceller, can be expressed as

$$r_{RF}(n) = o_R(n) * w_{FFe}(n) * \tilde{h}_{RF}(n) \tag{14}$$

where $w_{FFe}(n)$ is the impulse response of the ESF for the open-loop subsystem of the front-rear channel and $\tilde{h}_{RF}(n)$ is the inverse Fourier transform of the misadjustment function. Thus, the PSD of the residual echo can be expressed as

$$S_{r_{RF}}\left(e^{j\omega}\right) = S_{o_R}\left(e^{j\omega}\right) \cdot \left| W_{FFe}\left(e^{j\omega}\right) \tilde{H}_{RF}\left(e^{j\omega}\right) \right|^2 \tag{15}$$

which depends on the PSD of the output signal that will be played back through the rear loudspeakers of the reinforcement system, $S_{o_R}(e^{j\omega})$, and on the squared magnitude of the misadjustment function, $|\tilde{H}_{RF}(e^{j\omega})|^2$, along with the squared magnitude of the ESF of the open-loop subsystem of the front-rear channel, $|W_{FFe}(e^{j\omega})|^2$.

According to (15), we can express the squared magnitude of the product of the open-loop subsystem ESF of the front-rear channel and the misadjustment function as

$$\left| W_{FFe}\left(e^{j\omega}\right) \tilde{H}_{RF}\left(e^{j\omega}\right) \right|^2 = \frac{S_{r_{RF}}\left(e^{j\omega}\right)}{S_{o_R}\left(e^{j\omega}\right)} \tag{16}$$

The PSD of the rear output signal, according to Figure 15-2, can be expressed as

$$S_{o_R}\left(e^{j\omega}\right) = S_{e_R}\left(e^{j\omega}\right) \cdot K_R^2 \cdot \left| W_{RFe}\left(e^{j\omega}\right) W_{Rn}\left(e^{j\omega}\right) \right|^2 \tag{17}$$

and thus, combining (16) and (17) and substituting into (11), we can obtain the expression for the closed-loop ESF of the front-rear channel that responds to

$$W_{RF}\left(e^{j\omega}\right)=1-\sqrt{\frac{S_{r_{RF}}\left(e^{j\omega}\right)}{S_{e_R}\left(e^{j\omega}\right)}} \tag{18}$$

which depends on the PSD of the residual echo remaining after the closed-loop subsystem of the front-rear channel, $S_{r_{RF}}(e^{j\omega})$, and on the PSD of the error signal of the front-rear channel, $S_{e_R}(e^{j\omega})$.

In the same way, we can obtain the expression for the ESF for the closed-loop subsystem of the rear-front channel that must follow

$$W_{FR}\left(e^{j\omega}\right)=1-\sqrt{\frac{S_{r_{FR}}\left(e^{j\omega}\right)}{S_{e_F}\left(e^{j\omega}\right)}} \tag{19}$$

4. PERFORMANCE MEASURES

In this section, a performance evaluation of the residual echo suppression filters for the closed-loop subsystems is presented. For the evaluation, we used four different impulse responses corresponding to four different real electro-acoustic paths measured in a medium-size car with 600 coefficients each, using a sampling rate of 8 kHz.

The misadjustment between the impulse response of the electro-acoustic path and the impulse response of the corresponding adaptive filter is controlled by adding a random noise to each one of the coefficients of the original impulse response. This estimation error can be measured by using the normalized l_2 norm of the weight misadjustment vector defined as

$$\left\|\varepsilon\right\|^2=\frac{\sum_{k=0}^{L}\left|h'_k-\hat{h}'_k\right|^2}{\sum_{k=0}^{L}\left|h'_k\right|^2} \tag{20}$$

where h'_k is the kth coefficient of the impulse response of the real electro-acoustic path and $\hat{h}'_k$ is the kth coefficient of its corresponding adaptive filter.

Several noise free speech recordings were used as passenger's speech adding real car noise, recorded while driving on a highway, as background noise resulting in a SNR around 20 dB. The length of each signal frame was 16 ms and to reduce the overall delay of the system, a time overlap of 75\% was used.

In order to measure the benefit of using the ESF for the closed-loop subsystems, the isolation between channels is used. That is defined as the ratio between the power of the front-rear channel output and the power of the rear-front channel output when only the front passenger is talking.

$$I_{RF} = \frac{E\left[\left|o_R(n)\right|^2\right]}{E\left[\left|o_F(n)\right|^2\right]} \tag{21}$$

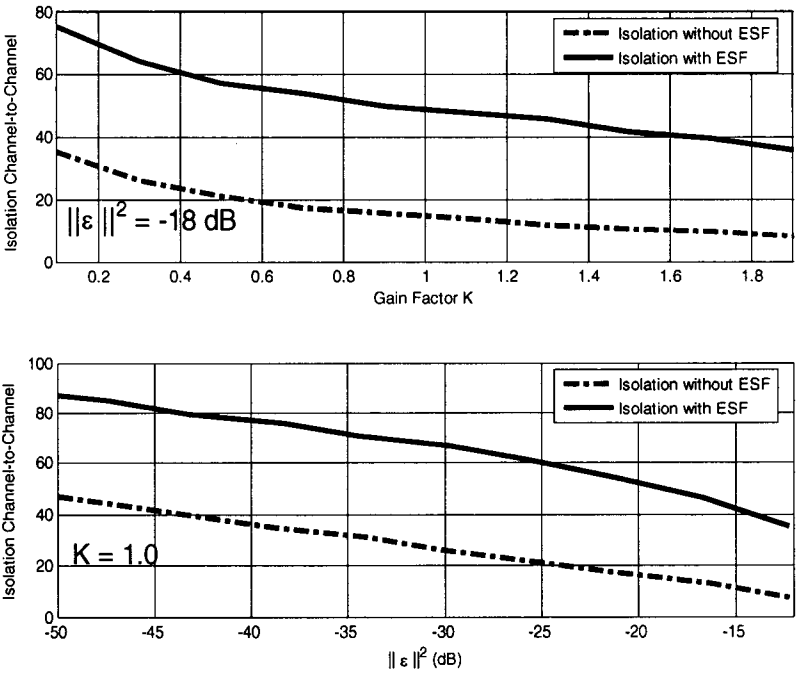


Figure 15-3. Isolation between channels with and without ESF in the closed-loop subsystems.

In the upper half of Figure 15-3, the evolution of the isolation between channels with the gain factor K for $\|\varepsilon\|^2 = -18$ dB is presented. It can be seen that the increase is around 40 dB for almost every value of K. The evolution of the isolation between channels with the normalized l_2 norm of the weight misadjustment vector is plotted below for $K = 1.0$. The isolation

increase ranges from 30 dB for high values of misadjustment (around -12 dB) to 40 dB for lower values of $\| \varepsilon \|^2$.

To shown that there is no degradation in terms of system gain decrease or distortion increase, the evolution of the system gain with K for $\| \varepsilon \|^2 = -18$ dB, and the evolution of the system gain with $\| \varepsilon \|^2$ for $K = 1.0$ is presented in Figure 15-4. In Figure 15-5, the evolution with K for $\| \varepsilon \|^2 = -18$ dB and with $\| \varepsilon \|^2$ for $K = 1.0$ of the Itakura-Saito distortion between the input signal and the corresponding output signal is presented.

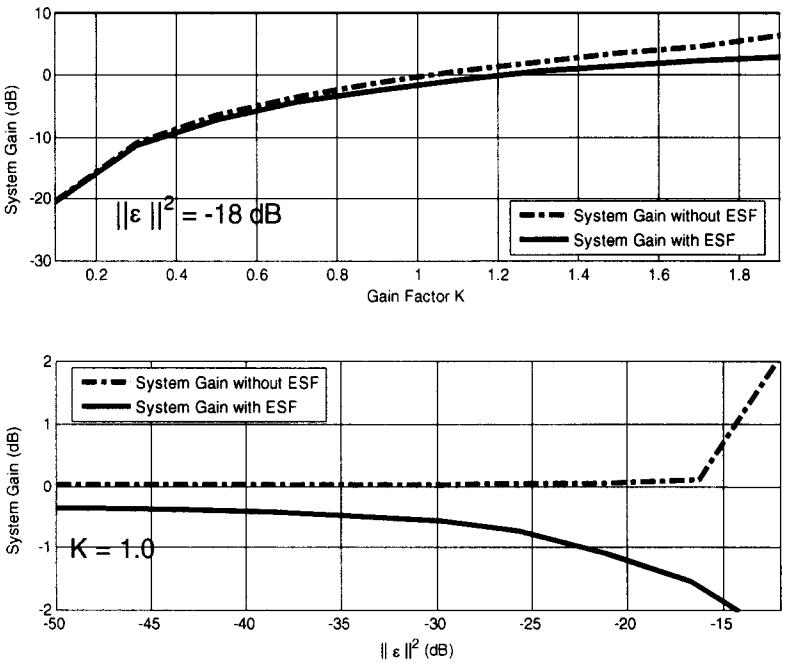


Figure 15-4. System gain with and without ESF in the closed-loop subsystems.

In the lower half of Figure 15-4 it can be seen that the System Gain increases dramatically for values of $\| \varepsilon \|^2$ above -15 dB. The same effect can be observed in both parts of Figure 15-5 regarding the distortion for high values of K or $\| \varepsilon \|^2$. This is due to the appearance of howling as the system is very close to instability and strong tonal components are present in the output signal.

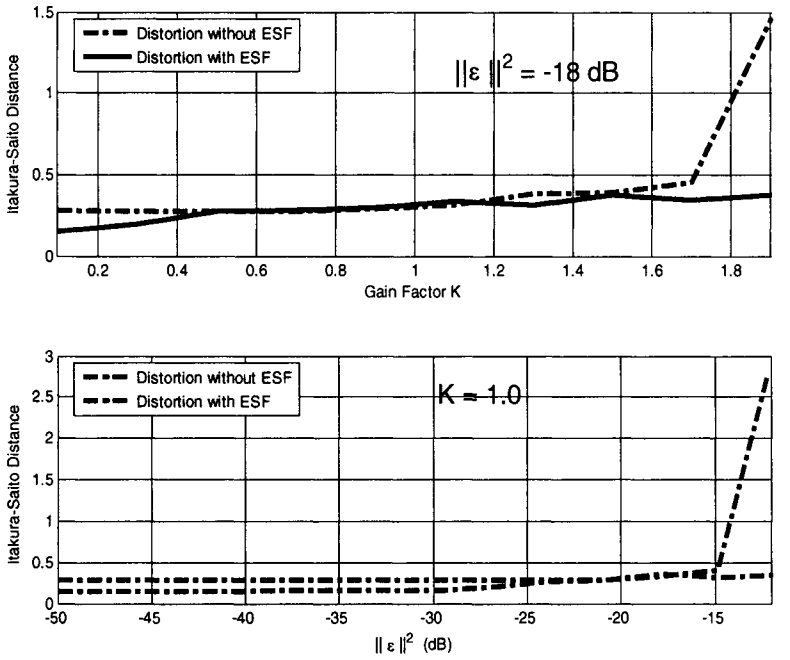


Figure 15-5. Itakura-Saito Distance between the input signal and the reinforced speech with and without ESF in the closed-loop subsystems.

5. CONCLUSION

A two-channel speech reinforcement system is required in order to make communications inside a car more comfortable. In a two-channel system, two subsystems can be distinguished for each channel, an open-loop and a closed-loop subsystem. The use of specific treatment for residual echo attenuation in the closed-loop subsystems has been presented, and the optimal expression for the transfer function of the Echo Suppression filter that ensures unconditional stability has been derived. Optimal echo suppression filters do not always exist and the existence of the optimal filters depends on the misadjustment function between the electro-acoustic path impulse response and the adaptive filter of the acoustic echo canceller which is not under the control of the designer. An alternative solution has been proposed and evaluated. This solution is based on an estimation of the residual echo power spectral density. The performance evaluations show that there is an increase of around 40 dB in the isolation between channels when using the proposed Echo Suppression Filters, without decreasing the gain of the system or increasing the speech distortion.

REFERENCES

[1] Breining, C., Dreiseitel, P., Hänsler, E., Mader, A., Nitsch, B., Puder, H. Schertler, T., Schmidt, G. and Tilp, J., 1999, Acoustic echo control. An application of very-high-order adaptive filters, IEEE Signal Processing Mag., vol. 16, no. 4, pp. 42-69, July 1999.

[2] Enzner, G., Martin, R. and Vary, P., 2002, Unbiased residual echo power estimation for hands-free telephony, in Proceedings of ICASSP, vol. 2, pp- 1893-1896. May 2002

[3] Gallego, F., Ortega, A., Lleida, E. and Masgrau, E., 2002, Method and system for suppressing echoes and noise in environments under variable acoustic and highly feedback conditions, patent WO 02/101728 A1.

[4] Gustafsson, S., Martin, R. and Vary, P., 1998, Combined acoustic echo control and noise reduction for hands-free telephony, Signal Processing, no. 64, pp. 21-32, January 1998.

[5] Hänsler, E., and Schmidt, G. U., 2000, Hands-free telephones - joint control of echo cancellation and postfiltering, Signal Processing, no. 80, pp. 2295-2305. January 2000.

[6] Ortega, A., Lleida, E. and Masgrau, E., 2003, Residual echo power estimation for speech reinforcement systems in vehicles, in Proceedings of Eurospeech. September 2003.

[7] Ortega, A., Lleida, E. and Masgrau, E., 2005, Speech reinforcement system for car cabin communications, IEEE Transactions on Speech and Audio Processing. vol. 13, no. 5. pp. 917-929. September 2005.

Chapter 16

NOISE SOURCE CONTRIBUTION OF ACCELERATING CARS AND SUBJECTIVE EVALUATIONS

Shunsuke Ishimitsu[1]
[1]*Department of Mechanical Engineering, University of Hyogo, Japan*

Abstract: Recently many researchers working in the field of time-frequency analysis using based on wavelet transform have focused on analyzing wavelets that are derived using a mathematical approach. In our proposed technique, a measured signal has been adopted as the wavelet, and we analyze the correlation between acoustic signals in the car cabin and suction noise signals. Because traditional calculations of correlation repeat the averaging procedure, the original signal must be stationary. Consequentially, a technique for separating and identifying noises from each part of the engine has been used for noise source contribution analysis. To apply the method to time-varying signals, the concept of an instantaneous correlation factor (ICF) is introduced, and we prove that a dominant feature of the correlation can be estimated by the ICF. The time-varying correlation for noise source contribution analysis of an accelerating car is analyzed. In addition, a fundamental experiment on audibility impressions in that case has been also conducted.

Keywords: Wavelet, car interior noise, correlation, subjective evaluation

1. INTRODUCTION

Frequency analysis based on Fourier transform is commonly used for extracting the features of audio signals. However, this analysis assumes that the signal being analyzed is periodic and stationary. In order to accurately represent general physical signals, it is necessary to analyze time characteristics as well as frequency characteristics; this is known as time-frequency (t-f) representation. Typical methods of analysis include the spectrogram, Wigner Distribution (WD) [1] and wavelet transform (WT) [2]. The t-f resolution features of WT is characterized by a multiple structure,

which has high frequency resolution in the low frequency range, and high time resolution in the high frequency range. This method, which is used in applications in a wide range of fields [3], performs analysis using the affine transformations (similarity transformations and translations) of a base function known as an analyzing wavelet (AW), whose distribution is localized in both time and frequency.

On the other hand, the suction noise is an important element in the engine sound during the acceleration. When considering the sound system design inside the car, the manufacturers have been working on modules to combat suction noise in the cabin.

In this chapter we have examined the correlation characteristics of non-stationary signals with the objective of identifying sound sources in the interior of an automobile. Instantaneous correlation functions (ICFs), *ICF(t,f),* and *ICF(t,a)* based on the WT analysis method [4],[5] were employed along with the time-time analysis (TT). First, the effectiveness of the ICF has been verified using simulation signals, and then the contribution of intake noise, a component of engine noise, to overall noise in the interior of a car was investigated during acceleration. Using an ICF where signals relating to each noise source are selected for the actual signal AW, we have demonstrated that this technique is also useful for contribution analysis. A fundamental experiment about audibility impressions was also conducted.

2. CAR INTERIOR NOISE AND CONTRIBUTION ANALYSIS

The elements which constitute in-car noise are the engine noise (intake noise, exhaust noise, etc.), the whoosh sound of the cars passing by, and the road noise. It is known that the sounds, which are conspicuous from auditory perspective, include the engine noise during acceleration, the road noise when driving on a rough road, and the whoosh of air when driving at high speeds [6]. As the vehicle moves, the contribution of each noise can be determined using a coherence function, however, acceleration is the time when the effect of engine noise becomes conspicuous, and under acceleration conditions it is almost impossible to obtain a coherence function. Therefore, the coherence function technique cannot be used for contribution analysis. Nevertheless, among the engine noise components present during acceleration, intake noise is a critical factor when we consider the timbre of the car interior noise, and it is thus important for manufacturers to be able to analyze the contribution of intake noise to the overall car interior noise. Since a coherence function cannot be obtained, and the

techniques currently in use is an extremely labor-intensive technique which involves isolating each contributor to the overall noise. Furthermore, although autoregressive vector modeling of a set of measurement signals has been applied to time-varying noise signals [7], this measurement is also highly labor-intensive.

In our proposed technique, we incorporate the ICF into the correlation analysis method to distinguish the contribution of intake noise and to confirm the effectiveness of this simple method.

3. INSTANTANEOUS CORRELATION FUNCTION (ICF)

3.1 Wavelet Transform (WT)

WT can be obtained by calculating the inner product of the signal f(t) and AW $\varphi_{a,b}(t)$ in the following formula:

$$WT_f(b,a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \varphi^*_{b,a} \left(\frac{t-b}{a}\right) f(t)dt \tag{1}$$

Here, a is the scale parameter and b is the shift parameter. These variables perform, respectively, a similarity transformation and the translation of $\varphi_{b,a}(t)$. The WT is originally expressed in the time-scale (t-s) plane, but it can be regarded as an approximation of the *t-f* distribution by using an AW which is localized in terms of time and frequency.

3.2 WT Using the Actual Signal as the Analyzing Wavelet

Correspondence with the scale parameter a in (1) is achieved by using a similarity transform of the actual signal (used as the AW), and correspondence with the shift parameter *b* is achieved by translating each similarity-transformed signal along the time axis. That is, when represented by an actual signal *s(t,a,b)* used as an AW, the affine transformation analysis for the similarity transform and translation is given by Formula (16.2) below. *s(t,a,b)* is defined as the actual signal wavelet.

$$C_f(b,a) = k_a \int_{-\infty}^{\infty} s(t,a,b) f(t) dt \tag{2}$$

where, k_a is a normalized constant, just like $1/\sqrt{a}$ in (2).

This method also enables the analysis using *s(t,a)* with the frequency components of the harmonic structure. For example, if a Gabor function [2] is used, then a wavelet for extracting a single frequency component (ω_p) will be needed. If the AW also has a structure like that shown in Figure 16-1, the technique can be applied to the analysis of observed signals which have a fundamental frequency and the harmonic components of that. Therefore, when analyzing a signal with a harmonic structure, features due to the fundamental frequency and its harmonic components have to be detected all together, and it is thus anticipated that changes in features with a harmonic structure can be identified.

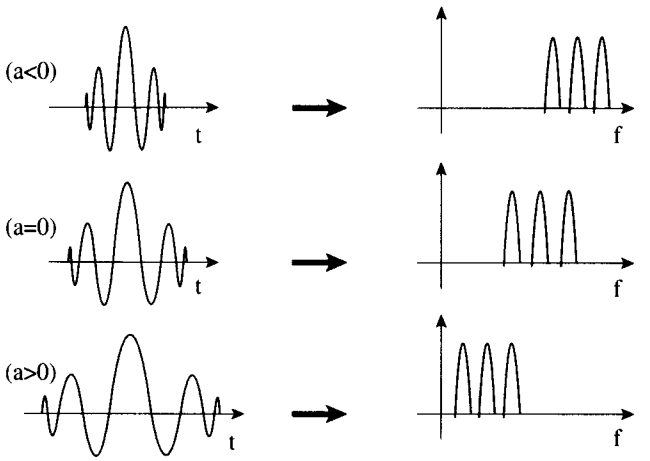


Figure 16-1. Harmonic AWs.

3.3 Definition of ICF

In the context of this study, the WT which uses *s(t,a)* as a base function is defined in accordance with (3) as the ICF *ICF(t,a)*:

$$ICF(t,a) = k_a \int_{-L_a/2}^{L_a/2} s(\tau,a) f(t+\tau) d\tau \tag{3}$$

where L_a is the window length determined by compression/extension. Depending on L_a, the value of *ICF(t,a)* varies, and it is normalized by k_a using an autocorrelation function. Calculation while shifting by τ is the same operation as using the shift parameter b in (1). If $a = 0$ in (3), a correlation function is applied. On the other hand, if $a \neq 0$, compression occurs if $a < 0$, and similarly, extension occurs if $a > 0$. If part of the observed signal turns out to be *s(t,a)*, then it is possible to analyze self-similarity between the AW and the observed signal in some instances when $a \neq 0$. In the analysis of two signals, it is possible to detect the similarity between *s(t,a)* removed from one signal and another different signal.

4. ANALYSIS OF A CHIRP SIGNAL

At first, we have considered a chirp signal. A basic AW signal can be cut at the center and is half its length. The analyzed result is illustrated in Figure 16-2. The vertical axis represents the length of the AW, which is varied in 2^n with respect to the basic AW length. In this case, as this analysis is an auto-correlation at the center of this observation time (80ms), the dilation and contraction rate is zero and the correlation value can be robust. At the end of this observation time, the values are shifted toward the construction. This part implies mutual-similarity analysis. So, the correlation analyses of the time-varying signals allow inclusion the mutual-similarity feature.

5. APPLICATION TO TIME-TIME ANALYSIS

As the influence of the ratio in degrees of Base AW in ICF is fixed, a new method was developed so that the time-varying composition analysis could be performed. By considering the case in which a time-varying AW, *g(t)*, is introduced into ICF as the Time-Time (TT) analysis [8]. If the relation between the acceleration time and rotation speed is known, the rpm-rpm expression can also be attained:

$$tt(t_1, t_2) = k_a \int_{t_2 - L_a/2}^{t_2 + L_a/2} g(t_2 + \tau) w_a(\tau) f(t_1 + \tau) d\tau \tag{4}$$

where w_a is an analysis window of length L_a, which changes according to the rotation speed. By this technique, the time-variation of a degree ingredient is introduced into ICF.

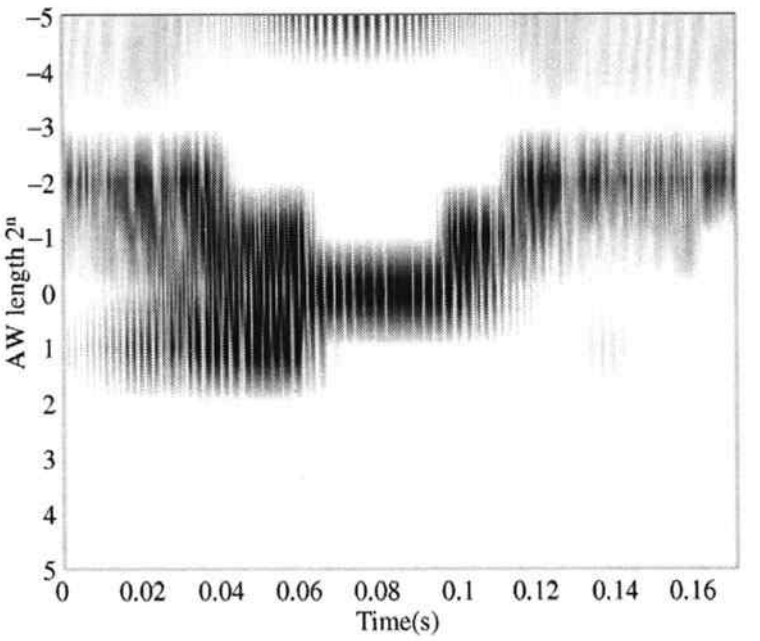


Figure 16-2. Analysis of a Chirp Signal.

6. ICF ANALYSIS OF CAR INTERIOR NOISE

6.1 Experiments

The vehicle used in the present study was a sedan with a left-side steering wheel and an in-line 4-cylinder engine. A dummy head was installed in the passenger seat, and analysis was carried out to determine the correlation between the signals obtained from eardrum microphones in the dummy head and the intake noise signals obtained near the intake duct. The engine running conditions were full acceleration at three different speeds. Engine rotation speed varied from 1000 rpm to 6000 rpm, and the acceleration period was approximately 20 seconds. With an in-line 4-cylinder engine, the engine fires twice in each rotation, generating a secondary vibration force, and these higher harmonics are the main component of noise. As noted earlier, engine noise is dominant during acceleration, and this noise (which has a harmonic structure) is the most conspicuous auditory component. Intake noise has a significant effect on this engine noise, and the correlation

between the intake noise signal and interior noise during transitions is therefore important.

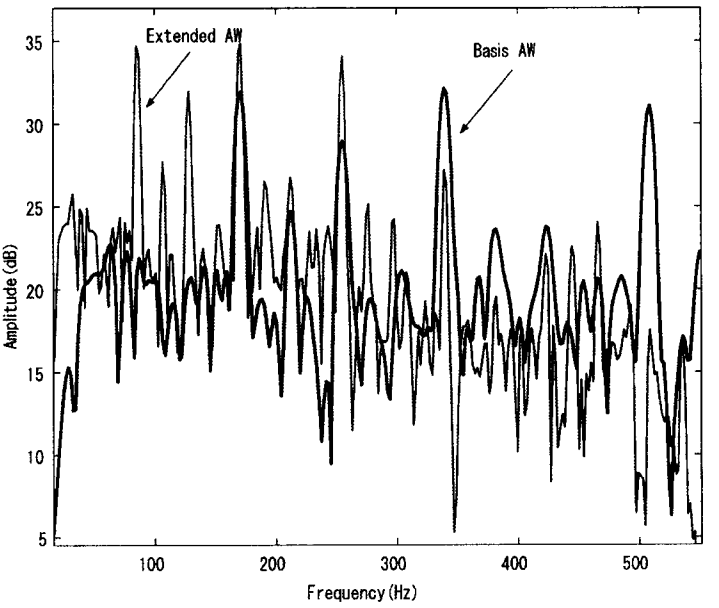


Figure 16-3. Dilation of a basis AW.

6.2 ICF Analysis of Car Interior Noise

An AW was selected to act as the basis for ICF analysis of car interior noise. The basis AW was created using intake noise data from normal running at both 2000rpm and 5000rpm as reference data, and an extra experiment was conducted. Better results were obtained using 5000rpm (which extends the basis AW) as the basis AW. Figure 16-3 shows the frequency characteristics when the basis AW is extended. The advantage of using the signal from normal running at 5000rpm as the basis is that the case in which the AW is extended corresponds to low-order analysis, so the analysis bandwidth is smaller and higher harmonics can easily be captured.

Figure 16-4 shows the results of ICF analysis of car interior noise during acceleration using this basis. Our results indicate that the effect of intake noise is great at low-speed rotation, and that the contribution gradually becomes weaker, but then increases when a high rotation speed is reached at the end. This result is expressed not by the components of each order, but as the total correlation value of their energy. In results of order analysis, it was

observed that the power of the 2nd order components is large near 1000-2000 rpm, and that 4th order components make a large contribution near 1000rpm and 5000 rpm. The ICF analyzes the degree of contribution of all harmonic components, and high values were obtained near 1200 rpm (taking ICF (1200, 3.8) to be the center) and near 5000 rpm for ICF (5000, 0). Our results are thus consistent with those previously acquired using conventional techniques. In. Figure 16-5, the same basis was used to perform contribution analysis together with noise near the exhaust manifold in the engine compartment. Figure 16-5 differs from Figure 16-4 in that a high value was obtained near 2000rpm (taking ICF (2000, 1.5) to be the center), implying that the constituent elements of engine noise contribute to the structure at each rotation speed. Figure 16-5 is a direct result of conducting TT analysis using (4), where *g(t)* is the suction noise and *f(t)* is the car interior noise. Such analysis is possible because the acceleration signal is synchronized. When taking into consideration the harmonic structure by Time-varying AW *g(t)*, a difference with ICF around 5000 rpm, and serves as a value slightly lower than the ICF value. This result shows that the power of the 4^{th} and the 6^{th} components is small around 5000 rpm. In addition, the energy component appearing near 1000rpm in the right-hand side of the figure is due to the influence of the noise.

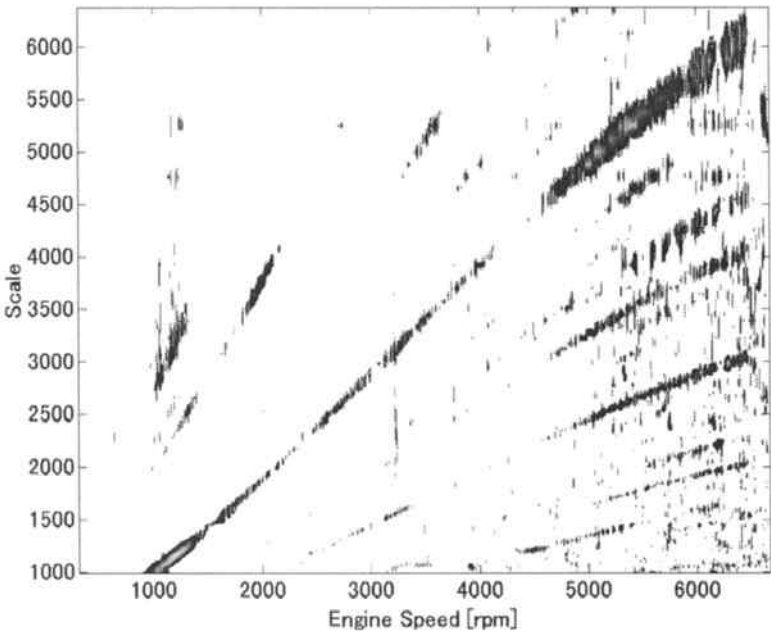


Figure 16-4. ICF of engine exhaust noise : ICF(t,a).

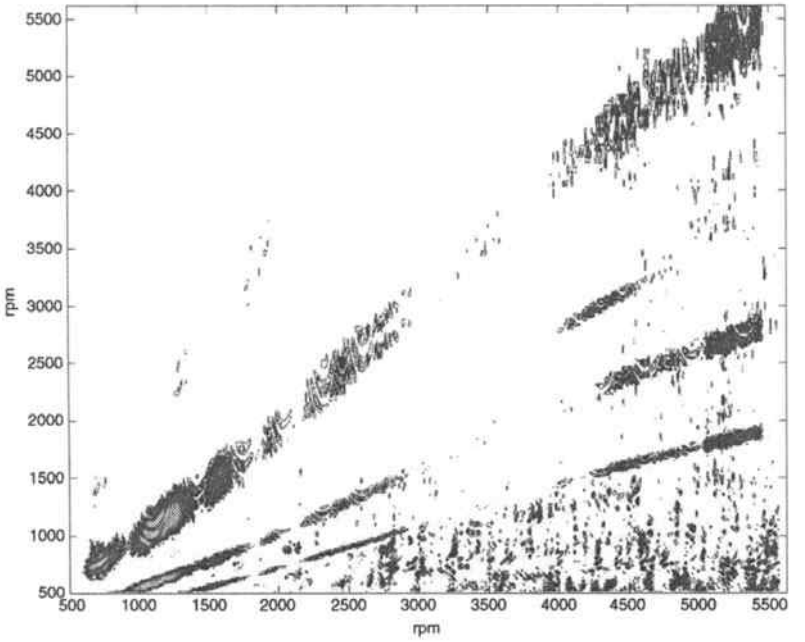


Figure 16-5. TT of car interior noise: TT(t1,t2).

Subjective evaluation:

Although it was explicitly shown that it is possible to analyze the time-varying correlation characteristic of suction noise, the correspondence between the time change and the audibility impression of suction noise remained unclear. Thus, the inclination of the time change on a t-f plane and amplitude were changed and an examination in which descriptive adjectives were selected by human subjects was performed.

Six types of sound sources were presented twice to 20 subjects (16 males; 4 females) of normal hearing aged 19 to 21, and they were asked to evaluate each. Evaluation was performed by subjective selection between 13 related pairs of adjective measures. The sound sources were as follows:

1. Acceleration sound of a full throttle.
2. Sample with 0.67 times inclination of the time change on a t-f plane of 1.
3. Sample with 1.5 times inclination of the time change on a t-f plane of 1.
4. Sample with one-half the amplitude of 1.
5. Sample with one-half the amplitude of 2.
6. Sample with one-half the amplitude of 3.

Among the data in which the adjective selections suggested a predominant difference, there was a tendency for a large time change on the t-f plane to incline toward the positive factor. Table 16-1 shows the correlation coefficients for the adjectives suggesting the significance of the frequency time rate of change and sound pressure levels.

It turns out that the pairs showing correlation with the t-f plane demonstrate an inclination toward an energy ingredient, such as "fast-slow", "alive-paralyzed", "sporty-not sporty", "dynamic-calm", "metallic - dull", "rough-smooth", "bright-dark" and "high-low".

Table 16-1. Correlations between adjectives.

Adjective pair	Correlation coefficient
fast - slow	0.796948
strong - weak	0.40073
alive - paralyzed	0.705082
sporty - not sporty	0.761852
metallic - dull	0.695459
pleasant - annoying	0.422132
dynamic - calm	0.748415
expensive - cheap	0.27309
luxurious - simple	0.286475
rough - smooth	0.767504
dark - bright	0.67921
high - low	0.749835

7. CONCLUSION

Primary goal of the present study was to analyze the correlation of non-stationary signals, for which purpose we introduced an ICF that uses actual measured signals as AWs. The interior noise of a sedan with an in-line 4-cylinder engine was analyzed using this technique. Among the components of engine noise (which is considered to be auditory-dominant during acceleration), we focused on intake noise, using the ICF to analyze the correlation of intake noise with interior noise during acceleration. Our results agreed with previously established data, and we thus conclude that the ICF may offer a method which could replace the present labor-intensive technique.

This work has resulted in a promising method for TT analysis. It is expected that this analysis can provide results nearer to the actual case since it can incorporate the time-varying structure of a degree ratio. However, there is still a lot of room for improvement. Moreover, subjective evaluation confirmed that the audibility impression of acceleration is dependent on the inclination of the time change on the time-frequency plane. However, since the differences were delicate, inclination and volume were not able to acquire a strong correlation. It is thought that there is room for further such examination.

In the future, we plan to analyze signals other than intake noise, and to continue developing our technique so that it can be used to clarify the component elements of engine noise during acceleration.

ACKNOWLEDGEMENT

The authors wish to express their deep appreciation to Mr. Kumano and Mr. Hatano of the Engineering Research Division of the Mazda Motor Corporation and H. Kobayashi working for the Olympus Corporation both in Japan for their cooperation and advice in carrying out this research.

REFERENCES

[1] T.A.C.M. Claasen and W.F.G. Mecklenbrauker, "The Wigner Distribution - A Tool for Time-Frequency Signal Analysis, part 1: Continuous-Time Signals", Phillips Journal of Research, vol. 35, no.3, pp. 217–250, (1980).

[2] I. Daubechies, Ten lectures on wavelets. SIAM, Philadelphia, 1992.

[3] O. Rioul and M. Vetterli, "Wavelets and Signal Processing", IEEE SP MAGAZINE, Vol.10, pp.14–38, (1991).

[4] S. Ishimitsu, H. Kitagawa, S. Horihata and N. Hagino, "Correlation Analysis of Time-varying Signal using Wavelets and its application to the ship interior noise analyses", JSME 69-682, C, pp.1529–1535, (2003).

[5] S. Ishimitsu, "Correlation Analysis of Ship Interior Noise using Wavelets", JSME No.985-2, pp.247–248, (1998).

[6] H. Hoshino and Y. Kozawa, "Evaluation of Sound Components of Passenger Car Interior Noise", R\&D Review of Toyota CRDL, Vol.30, No.3, pp.29–38, (1995).

[7] H.D.V. D. Auweraer, L. Hermans, D. Otte and M. Klopotek, "Time Dependent Correlation Analysis of Truck Pass-By Noise Signals", SAE Technical Papers, D.N.971986, pp.915–922, (1997).

[8] S. Ishimitsu, "Correlation Analyses of Time-varying Signal using Wavelets and its application to the car interior noise analyses", JSME, 2004 Annual conference Proceedings V, pp.273–274, (2004).

Chapter 17

STUDY ON EFFECT OF SPEAKER VARIABILITY AND DRIVING CONDITIONS ON THE PERFORMANCE OF AN ASR ENGINE INSIDE A VEHICLE

Shubha Kadambe
Signal and Image Processing, Office of Naval Research and University of Maryland, College Park, USA.. E-mail: Shubha_Kadambe@onr.navy.mil

Abstract: Spoken dialogue based information retrieval systems are being used inside vehicles. The user satisfaction of using such a system depends on how an ASR engine performs. However, the performance of an ASR is affected by speaker variability, driving conditions, etc. In this chapter, we report the study that we performed to analyze these effects of speaker variability, different driving conditions and the effect of driving task on the ASR performance. This study consists of experimental design, data collection and systematically testing an ASR engine using this data. From the obtained results, it can be observed that (I) the ASR performance exhibits (a) significant speaker variability since the stress of driving task varies from speaker to speaker, (b) significant performance degradation across driving conditions since the noise type and level varies and (c) significant effect of driving task on recognition performance, and (II) the effect of live noise on recognition performance is not same as adding car noise to the pre-recorded speech data. The former observations are important since by just training an ASR engine on lots of speech data will not help and it is essential to include stress factors and cognition load in ASR engines to improve its performance.

Key words: Driving task, driver work load, ASR performance, speaker variability, effects of driving conditions on ASR performance.

1. INTRODUCTION

Spoken dialogue information retrieval applications are becoming popular in automobiles. Due to the typical presence of background noise and multi-tasking (i.e., speaking while driving) the speech recognition accuracy is affected significantly. This in turn affects the overall performance of the information retrieval system and the user's satisfaction. Hence, there is a need for a systematic study of the effect of speaker variability, different driving conditions and the driving task on the performance of an ASR engine which would help in improving the user's satisfaction. In [1], the effect of speech recognition performance on driving task of a driver is studied. However, in this chapter, the effect of driving task on recognition performance is studied. To the knowledge of this author such a study has not been reported before. The study in [1] indicates that the recognition accuracy significantly affected the driving task across drivers of different ages. The study reported in this paper helps in understanding how the speech of a driver is affected by the stress of the driving task which in turn affects the ASR performance. By understanding this effect and noting from [1] that by improving the ASR accuracy driving task can be improved, it is hoped that the ASR performance can be improved by including the stress and driver's work load while training an ASR system. This results in improved driving task and thus the driver's safety. The rest of the paper is organized as follows. In the next section, the experimental design in terms of estimating the required sample size that is needed to perform the proposed study is described. In section 3 the details of data collection is provided. Section 4 provides the results of testing an ASR engine using the collected data. Finally, in section 5, we conclude and discuss the future work.

2. EXPERIMENTAL DESIGN

The experiment of studying the effect of speaker variability driving conditions and driving task on the ASR performance is designed first by estimating the minimum sample size that is needed to collect enough data that provides statistically significant results. For this sample size estimation, we use word error rate (WER) of an ASR engine as a measure to compare the different datasets.

For the sample size computation the following equation is used:

$$\Phi^2 = \frac{s' \frac{\left(\sum_i (\mu_i - \mu)^2\right)}{a}}{\sigma^2} \quad (1)$$

where Φ is a parameter that relates to statistical power i.e., the probability of correctly rejecting the null hypothesis, a is the number of experiments, μ_i the expected average mean WER for each experiment (or dataset), μ is the overall average WER, σ is the pooled within groups error standard deviation and s' is the estimated sample size.

We assume that the mean WER for each experiment varies from the overall mean by 5 for all datasets by keeping in mind a difference of 3-4 % is statistically significant and the standard deviation is assumed to be 10. We use $\alpha = 0.05$. By substituting these values in the above equation we get:

$$\Phi^2 = \frac{s'}{4} \;\&\; \Phi = \frac{\sqrt{s'}}{2}. \quad (2)$$

Using the curves plotted in [2] for power versus Φ, for $\alpha = 0.05$, degrees of freedom for numerator = $(a - 1) = 2$, ($df_{denominator}$ = infinity) for a power of 0.8 which corresponds to $\Phi = 3$ we get an estimated sample size $s' = 36$. Note that a is selected as three as we conduct three different experiments. So the minimum sample size of 40 for each experiment (five sentences/per speaker and total of 8 speakers) seems to have good enough statistical power.

3. DATA COLLECTION

Based on the above described sample size estimation, we decided to collect data from twelve speakers with each speaker uttering twenty sentences. Note that this sample size is more than the estimated sample size and hence, our ASR performance results should be statistically significant. We chose all twelve speakers as male to avoid the effect of inter i.e., male versus female speaker variability. Since our overall goal is to improve the user's satisfaction of using the information retrieval systems inside a vehicle, we selected twenty utterances related to queries on weather information at different cities. These twenty sentences are listed in Table 17-1. The driving conditions under which data was collected are mentioned in Table 17-2. The following datasets were collected using 12 male subjects.

1. Data set1: Selected sentences spoken by each subject in a vehicle (SUV) under each driving condition mentioned in Table 17-2.
2. Data set2: Recordings of selected sentences spoken by the subjects inside the vehicle when it was stationary. This is equivalent to data collected in a laboratory.
3. Data set3: Playback data of the recorded sentences (data set2) inside a vehicle under different driving conditions mentioned in Table 17-2.

Table 17-1. Selected weather related speech utterance.

1	I want to know if it will rain in Seattle on Sunday
2	What will the weather be like in Phoenix?
3	Can you repeat that?
4	How is the weather in Hartford, Connecticut?
5	I'm looking for the extended forecast for Houston
6	What is the forecast this weekend for Buffalo?
7	What about Saturday?
8	Please tell me the weather today in Portland, Oregon?
9	Thank you
10	What cities do you know in California?
11	How is the weather in Miami?
12	Please tell me what the weather will be like tomorrow in New York City?
13	Can you give me the forecast for Boston?
14	Is it raining in Pittsburgh?
15	What was the high today in Pasadena?
16	Will it rain in Denver today?
17	What is the current temperature in Fargo, North Dakota?

Table 17-2. Conditions under which sample speech data were collected.

Test Condition	**1.**	**2.**
Road Surface 4 (Freeway, 50 – 60 MPH)	•	•
Windows Up	•	•
Windows Down		
Fan Off	•	
Fan On		◎
Tape (radio) Off	•	
Tape (radio) On		◎
Turn Signal Off	•	
Turn Signal On		◎
Windshield Wipers Off	•	
Windshield Wipers On		◎
Comments	Baseline on freeway("quiet")	Combined effect("noisy")

These three datasets were collected using two microphone arrays and one clip microphone which was clipped to a shirt in front below the chin of a subject. The two microphone arrays are: 1. CSLR [3] and 2. Andrea. The microphone array built by CSLR has five microphones with five output channels where as Andrea microphone array has only one output channel. Seven channels – 5 channels of CSLR array, 1 channel of Andrea and 1 channel of clip microphone of data was recorded using an 8-channel DAT recorder from Fostex.

Note that the conditions 1 and 2 in Table 2 are referred as "quiet" and "noisy" in recognition accuracy tables and in figures in the next section whereas "stat" correspond to the data that was collected when the vehicle was stationary (data set2).

4. SIMULATION DETAILS AND ASR RESULTS

Recognition experiments were conducted using the three datasets mentioned above using only clip microphone channel data since the speech data quality from microphone arrays were poor and recognition accuracies were bad. The ASR engine that is used in this study is a continuous speech recognizer. The vocabulary size of this engine is 2000 words.

Table 17-3. Recognition performance in terms of word recognition accuracy in percentage across 12 subjects and across 3 driving conditions for live speech data collected in a vehicle.

Subject	Stat	quiet	Noisy	μ:across conditions/ subject	σ:across conditions/subject
1	92	86.9	84.8	87.9	3.703
2	95.7	95.6	95.5	95.6	0.1
3	83.8	82.7	78.4	81.63	2.854
4	94.9	93.4	89.8	92.7	2.621
5	91.1	89.7	84.9	88.57	3.252
6	95.6	78.8	73.9	82.77	11.38
7	98.4	94.5	90.6	94.5	3.9
8	98.4	81.2	70.9	83.5	13.89
9	98.4	90.6	87.4	92.13	5.658
10	89.8	87.6	87.4	88.27	1.332
11	94.1	85.6	75.8	85.17	9.158
12	95.8	95	90	93.6	3.143
μ: across subjects	94	88.47	83.82		
σ:across subjects	4.2819	5.6513	7.9392		

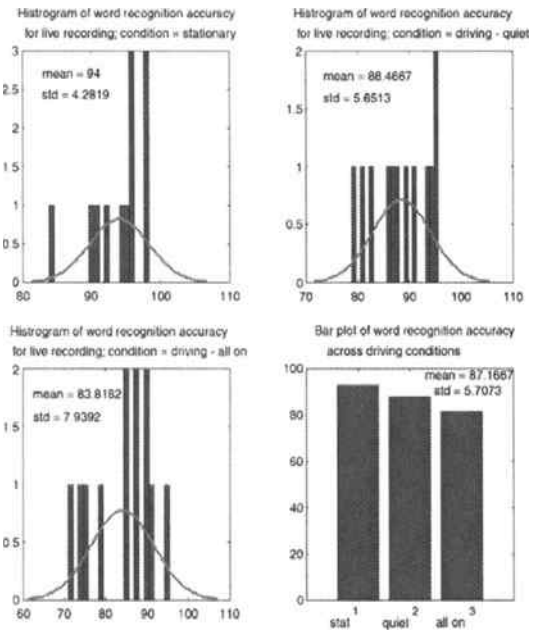


Figure 17-1. The histogram and bar plots of word recognition accuracy for live speech data collected in a vehicle for different driving conditions.

In Table 17-3, the performance of the ASR in terms of percentage of word recognition accuracy is provided for each subject under "stat", "quiet" and "noisy" conditions (data set1 and data set2). In addition, the mean and the standard deviation of word recognition accuracy across conditions for each subject are provided. The recognition accuracy across three conditions that was computed by pooling the speech data of all subjects under each condition is tabulated in Table 17-4. In Figure 17-1, the histogram plots for the recognition accuracy for these three conditions are provided. In addition, the bar plot of recognition accuracy across conditions is provided. From this bar plot and Table 17-4, it can be seen that the recognition accuracy degrades by 5 % for "quiet" as compared to "stat" and by 11 % for "noisy" as compared to "stat".

Table 17-4. Recognition performance in terms of word recognition accuracy in percentage for live speech data collected in a vehicle across conditions by pooling all data for each condition.

Stat	Quiet	Noisy	μ :across conditions	σ:across conditions
92.7	87.5	81.3	87.17	5.7073

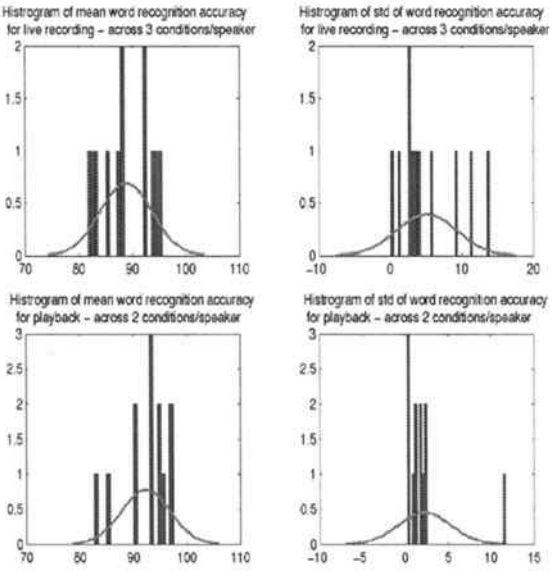


Figure 17-2. histogram plots of word recognition accuracy across conditions by pooling all subjects' data for live speech (row 1) and for play back speech (row 2).

In Figure 17-2 row 1, the histograms of mean and standard deviation across conditions for each subject are plotted. From Table 17-3, Figure 17-1 and Figure 17-2 row 1, it can be seen that the speaker variability across conditions is very significant.

Table 17-5. ASR's performance in terms of percentage of word recognition accuracy.

Subject	Quiet	noisy	μ:across conditions/subject	σ:across conditions/subject
1	94.1	92.3	93.2	1.2728
2	96.6	93.3	94.95	2.3335
3	83.9	81.5	82.7	1.6971
4	94.1	93.4	93.75	0.4950
5	91.1	89.7	90.4	0.9899
6	93.2	92.9	93.05	0.2121
7	97.6	96.9	97.25	0.4950
8	98.4	81.9	90.15	11.6673
9	96.9	93.7	95.3	2.2627
10	86.6	83.5	85.05	2.1920
11	95.8	93.2	94.5	1.8385
12	98.3	96.6	97.45	1.2021
μ:across subjects	93.88	90.74		
σ:across subjects	4.6211	5.4343		

In Table 17-5, the ASR's performance in terms of percentage of word recognition accuracy is provided for each subject under "quiet" and "noisy" conditions for play back data (data set3). In addition, the mean and the standard deviation of word recognition accuracy across these two conditions for each subject are provided. The recognition accuracy across two conditions that was computed by pooling the play back speech data of all subjects under each condition is tabulated in Table 17-6.

Table 17-6. Recognition performance in word recognition accuracy in percentage for play back speech.

Quiet	Noisy	μ: across conditions	σ:across conditions
93.0	89.0	91.0	2.8284

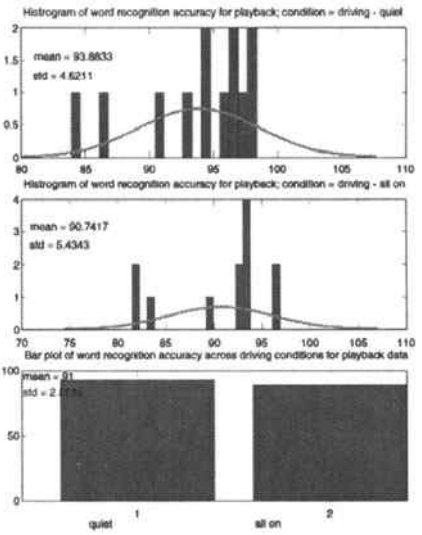


Figure 17-3. The histogram and bar plots of word recognition accuracy for play back speech data collected in a vehicle for different driving conditions.

In Figure 17-3, the histogram plots of recognition accuracy for the "quiet" and "noisy" conditions are provided. Further, the bar plot of the recognition accuracy across these two conditions is also plotted in Figure 17-3. From these, as in the live speech data case, it can be seen that the speaker variability across conditions is very significant. *Further, comparing Tables 17-3 and 17-5, it can be seen that the recognition accuracy is better for play back data for both "quiet" and "noisy" conditions indicating the*

effect of driving task on recognition performance. This difference in performance also indicates that live noise effects are not same as adding car noise to the pre-recorded speech data. In Figure 17-2 row 2, the histograms of mean and standard deviation across conditions for each subject for play back data are plotted.

In summary, the data analysis indicates:

- Significant speaker variability
- Significant performance degradation across conditions
- Effect of driving task on recognition performance
- Effect of live noise on recognition performance is not same as adding car noise to the pre-recorded speech data.

5. CONCLUSIONS

In this paper, as part of user satisfaction in using information retrieval systems in vehicles a systematic study on various factors that affect the performance of an ASR engine which is an integral part of an information retrieval system is reported. The sample size that provides required statistical power is estimated. While collecting data this estimated sample size is considered. Three different datasets were collected and analyzed using a continuous speech recognition engine. From the results it can be seen that, that (I) the ASR performance exhibits (a) significant speaker variability since the stress of driving task varies from speaker to speaker, (b) significant performance degradation across different conditions since the noise type and level varies and (c) significant effect of driving task on recognition performance, and (II) the effect of live noise on recognition performance is not same as adding car noise to the pre-recorded speech data. The latter observation is important since generally noise effect is studied by adding noise to clean speech. Former observations indicate that to improve the ASR performance and thus driver's safety, it is important to consider the effect of driving task on speech in terms of stress and Lombard effects which vary from speaker to speaker. It is also equally important to enhance speech quality by applying source separation techniques. Our future work will consider adding speech features that relate to stress and Lombard effect in our ASR engine and will apply HRL's blind source separation technique [3] to enhance speech data to further improve the recognition accuracy that is reported in [4].

REFERENCES

[1] A. W. Gellatly, "The use of speech recognition technology in automotive applications," PhD dissertation, Virginia Polytechnique Institute and State university, 1997.

[2] G. Keppel, *Design and Analysis: A researcher's handbook,* Prentice Hall Inc., Englewood Cliffs, NJ, 1973.

[3] M. Peterson and S. Kadambe, "A probabilistic approach for Blind Source Separation of underdertmined convolutive mixtures," in Proc. Of ICASSP, pp. VI-581-583, Hong Hong, April 6-10, 2003.

[4] "Robust ASR inside a vehicle using using blind probabilistic based under-determined convolutive mixture separation technique", In *DSP in Mobile and Vehicular Systems*, H. Abut, K. Takeda and J. H.L. Hansen (Editors), Springer Science, May 2005.

Chapter 18

TOWARDS ROBUST SPOKEN DIALOGUE SYSTEMS USING LARGE-SCALE IN-CAR SPEECH CORPUS

Yukiko Yamaguchi[1], Keita Hayashi[1], Takahiro Ono[1],
Shingo Kato[1], Yuki Irie[1], Tomohiro Ohno[1], Hiroya Murao[2],
Shigeki Matsubara[1], Nobuo Kawaguchi[1], Kazuya Takeda[1]
[1]Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan; [2]SANYO Electric Co. Ltd., 1-18-13 Hashiridani, Hirakata-shi, Osaka, 573-8534, Japan

Abstract: Researchers of the CIAIR project at Nagoya University have constructed a data collection vehicle and have collected about 179 hours of multi-modal data. Speech data from about 800 subjects have been transcribed and speech intentions, dependency structures, and dialogue structures to the text data have been annotated. Various research activities within the project's scope are continuing using the annotated data such as speech intention understanding and speaker's knowledge acquisition. In this chapter, we introduce these research activities and present the several findings from the in-car speech corpus.

Key words: Spoken dialogue corpus; spoken dialogue system; speech intention; dependency structure; dialogue structure;

1. INTRODUCTION

With the recent advances in continuous speech recognition technology, a considerable number of studies have been conducted on spoken dialogue systems. These have resulted in the collection of many large-scale corpora[1], and in turn, the active development of corpus-based systems. However, to use a corpus effectively in system development it is preferable that the corpus holds additional language information besides speech data and the transcribed text.

To improve the usefulness of the CLAIR corpus we have annotated speech intentions, dependency structures, and the associated dialogue structures. Using this annotated corpus we have been studying speech-intention understanding and extraction of information, and developing practical spoken dialogue systems. This paper introduces our research activities.

2. CIAIR IN-CAR SPEECH CORPUS

The Center for Integrated Acoustic Information Research (CIAIR) at Nagoya University has collected a large-scale corpus of in-car speech[2-4]. During the project, CIAIR members have configured a Data Collection Vehicle (DCV), shown in Figure 18-1, and collected about 400 GB of data by recording the spoken dialogues from 812 drivers. While the drivers were actually driving the DCV they carried out three dialogue sessions: with a human operator, with a WOZ system, and with a spoken dialogue system.

The collected speech data have been manually transcribed into ASCII text files according to the rules of the Corpus of Spoken Japanese (CSJ)[1]. An example of a transcript is shown in Figure 18-2. Each utterance is divided into utterance units separated by a pause of 200 ms or more. The transcribed text contains tags for grammatically ill-formed linguistic phenomena such as fillers, hesitations, and so on.

On investigating the features of the CIAIR corpus[5], we found that for the drivers' utterances the number per utterance unit of fillers, hesitations, and slips is 0.31, 0.06, and 0.03, respectively. Furthermore, we noticed that the drivers' utterances are affected by driving conditions, the gas and brake operation, and the steering-wheel operation.

In the studies introduced in this paper, we use restaurant guide dialogues between drivers and operators and between drivers and the WOZ system.

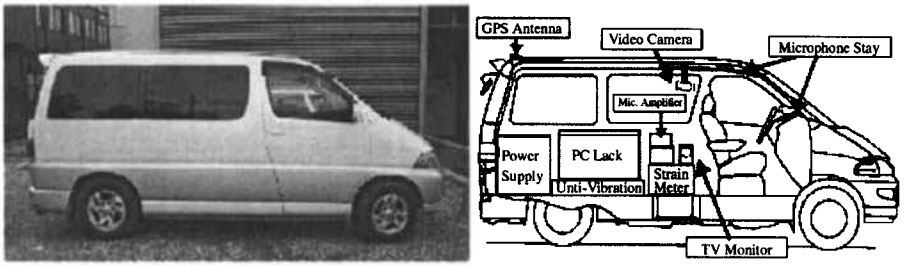


Figure 18-1. Data Collection Vehicle.

0028 - 02:21:353-02:24:909 F:D:II:			
(F えっと)ラーメンが	&	(F エット)ラーメンガ	[(Well,) chinese noodle]
ちょっと	&	チョット	[somehow]
食べたいんだけど	&	タベタインダケド	[want to eat]
どっか	&	ドッカ	[any restaurant]
ないかしら<SB>	&	ナイカシラ<SB>	[are there?]
0029 - 02:26:691-02:32:162 F:O:II:			
はい	&	ハイ	[Yes,]
この	&	コノ	[here]
近くですと	&	チカクデスト	[near]
該当する	&	ガイトースル	[suitable]
お店が	&	オミセガ	[restaurants]
三軒	&	サンケン	[three]
ございます<SB>	&	ゴザイマス<SB>	[there exist]

Figure 18-2. Example of transcription.

3. ELABORATION OF IN-CAR SPEECH CORPUS

3.1 Speech Intention Annotation

To gain a robust understanding of in-car spoken dialogues, we have designed layered intention tags (LITs)[6]. The intention tag expresses the task-dependent intention of the speaker, such as, request for a search, a statement of the search result, and so on. Each LIT is composed of four layers, "Discourse act," "Action," "Object," and "Argument." Figure 18-3 illustrates a part of a LIT's organization, showing that a lower-layered intention tag depends on the one above.

We have tagged over 35,000 utterance units manually and built the spoken dialogue corpus, which comprises 3,641 conversations in 1,256 sessions (Table 18-1), of LITs. Figure 18-4 shows a sample of a restaurant guide dialogue with LITs. The corpus contains 95 types of LIT.

Table 18-1. Speech intention annotated corpus.

Item	Size
Number of dialogues	3,641
Number of utterance units	35,421
Driver's utterances	16,224
Operator's utterances	19,187
Number of LIT types	95

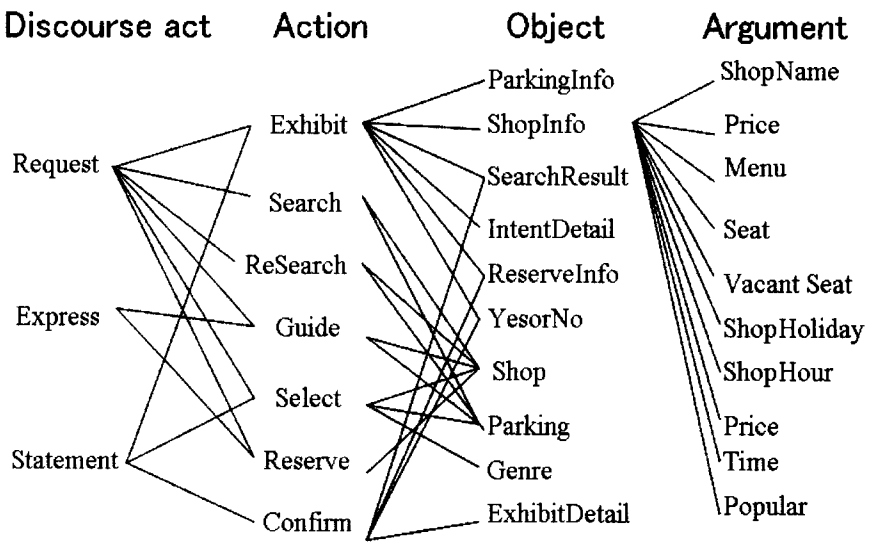


Figure 18-3. Layered Intention Tag (LIT).

#	Speaker	Utterance	LIT
0028	D	(えっと)ラーメンがちょっと食べたいんだけどどっかないかしら I want to eat Chinese noodle. Are there any restaurants?	Request + Search + Shop
0029	O	はいこの近くですと該当するお店が三軒ございます Yes, There are three restaurant near here.	Statement + Exhibit + SearchResult +ShopName
0030	O	ラーメンギョーザの金龍ラーメンさっぽろ亭春帆亭でございます Kinryu-Ramen of Chinese noodles and dumplings, Sapporo-tei, Shunban-tei.	Statement + Exhibit + SearchResult +ShopName
0031	D	さっぽろ亭でお願いします I'd like to go to Sapporo-tei.	Statement + Select + Shop
0032	O	はいさっぽろ亭ですと駐車場がございません Sapporo-tei has no parking lot.	Statement + Exhibit + ShopInfo + Parking
0033	O	よろしかったでしょうか Is it all right?	Request + Exhibit + YesorNo
0034	D	じゃ駐車場があるところお願いします Well, please let me know the restaurant with a parking lot.	Request + Re-search + Shop
0035	O	はい金龍ラーメンと春帆亭ですと駐車場がございますが Kinryu-Ramen and Shunban-tei have parking lots.	Statement + Exhibit + SearchResult +ShopName
0036	D	じゃ金龍ラーメンお願いします Well, I'll go to Kinryu-Ramen.	Statement + Select + Shop
0037	O	はいそれでは金龍ラーメンへご案内いたします Then, I'll guide you to Kinryu-Ramen.	Exhibit + Guide + Shop
0038	D	はい Thank you.	Statement + Exhibit + YesorNo

Figure 18-4. Example of a dialogue with LIT.

To evaluate the reliability of the LIT system, we conducted experiments using several annotators and evaluated the results with Cohen's kappa value. The κ value between the designers of LIT was 0.843 which exhibited good reliability. Even that between untrained annotators was 0.689 which indicated usable quality. Since trained annotators annotated the corpus, we confirmed that the corpus data were indeed reliable.

3.2 Dependency Structure Annotation

To characterize spontaneous dialogue speeches from the viewpoint of dependency, we have constructed a syntactically annotated spoken language corpus by providing morphological and syntactic information for each of the driver's utterances in the CIAIR in-car speech dialogue corpus[7].

We have provided boundaries between words, pronunciation, basic form, part-of-speech, conjugation type, and conjugated form of each word as the morphological information, and boundaries and dependencies between bunsetsus[6] as the syntactical information. In Japanese a dependency is a modification relation where a dependent bunsetsu depends on a head bunsetsu. That is, a dependent bunsetsu and a head bunsetsu work as a modifier and a modifyee, respectively.

Figure 18-5 shows the dependency structure of utterance No. 0028 in Figure 18-4, illustrating a sequence of bunsetsus. Each arrow represents a dependency relation between bunsetsus.

We have annotated the corpus by providing morphological and syntactic information for about 14,000 utterances. The corpus contains more than 85,000 morphemes and over 45,000 dependency relations.

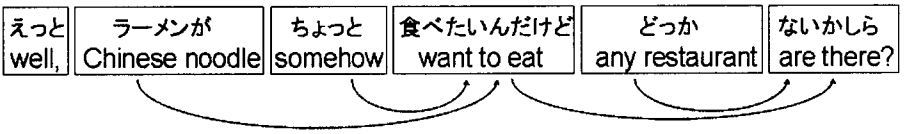


Figure 18-5. Dependency structure of utterance No.0028 in Fig. 18-4.

3.3 Dialogue Structure Annotation

To represent the structure of a dialogue, we consider a dialogue as a sequence of LITs and as such have provided structural trees[8]. If the dialogue-

[6] A *bunsetsu* is a linguistic unit used in Japanese, and roughly corresponds to a basic phrase in English. A bunsetsu consists of one independent word and more than zero ancillary words.

structural rules are obtained from the structural trees, it would be possible to apply usual language parsing technologies to analyze the dialogues. In this research, we used the first to third layers of each LIT and extended LIT with a speaker symbol like "D+Request+Search+Shop."

To construct a structural tree, we defined a category called POD (Part-Of-Dialogue), according to observations of the restaurant guide dialogue. The POD represents a partial structure of the dialogue, and Table 18-2 shows 11 types of POD.

We provided structural trees for 789 dialogues and obtained 297dialogue-structural rules. Figure 18-6 presents the dialogue-structural tree of the dialogue in Figure 18-4.

Table 18-2. POD (Part-Of-Dialogue) and its details.

POD	Detail	POD	detail
Guide	Guide	srch_rqst	Search request
Srch	Search	rsrv_rqst	Reserve request
p_srch	Parking search	s_info	Shop info
Slct	Select	p_info	Parking info
Genre	Genre	rsrv_dtl	Reserve details
Rsrv	Reserve		

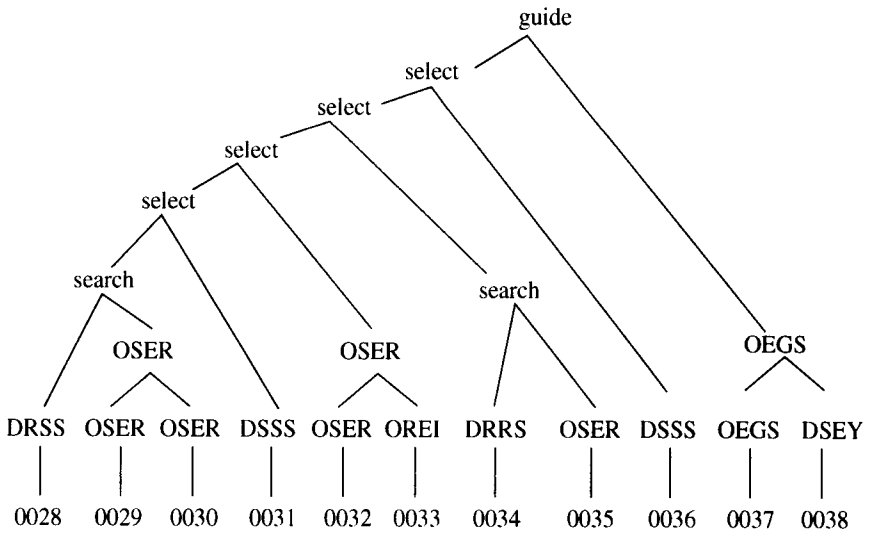


Figure 18-6. Dialogue-structural tree for the dialogue of Fig. 18-4.

In Figure 18-6 each LIT is represented by initials such DRSS. Because we express the dialogue structure of the restaurant guide dialogues, the root of the dialogue-structural tree is the "guide" POD.

4. ANALYSIS AND UTILIZATION OF IN-CAR SPOKEN DIALOGUE

4.1 Speech Intention Understanding using Decision Tree Learning

To determine the intention of an utterance, we constructed 32 decision trees[9] by applying the intention-annotated spoken dialogue corpus with LITs. As a set of attributes, we used the present speaker, the previous LIT (from first to third layers), the previous speaker, and the morphemes' appearance.

Figure 18-7 shows 32 decision trees. The inference algorithm using these decision trees is as follows:

1. The four trees in the first group are used, and the tree with the lowest re-classification error rate (the lowest error rate in the training data) is chosen.
2. The tag obtained in the first step (for example, the "Action" layer) is added to the attribute set, and the three trees in II of the second group are used. Then, one of the three trees is selected just as in the first step.
3. The tag obtained in the second step (for example, the "Discourse act" layer) is added to the attribute set and the two trees in A in the third group are used.
4. One tree in the fourth group is used for inferring the remaining undecided layer.

In our experiment we used 1,218 dialogues in the corpus with LITs, and divided 3,143 drivers' utterances into two groups. One is the training data, which consists of 2,972 utterances, and the other is the evaluation that comprises 171 utterances. We used See5[7] for decision-tree learning. The result of the evaluation experiment was a detection precision of 73.1%.

[7] http://www.rulequest.com/see5-info.html

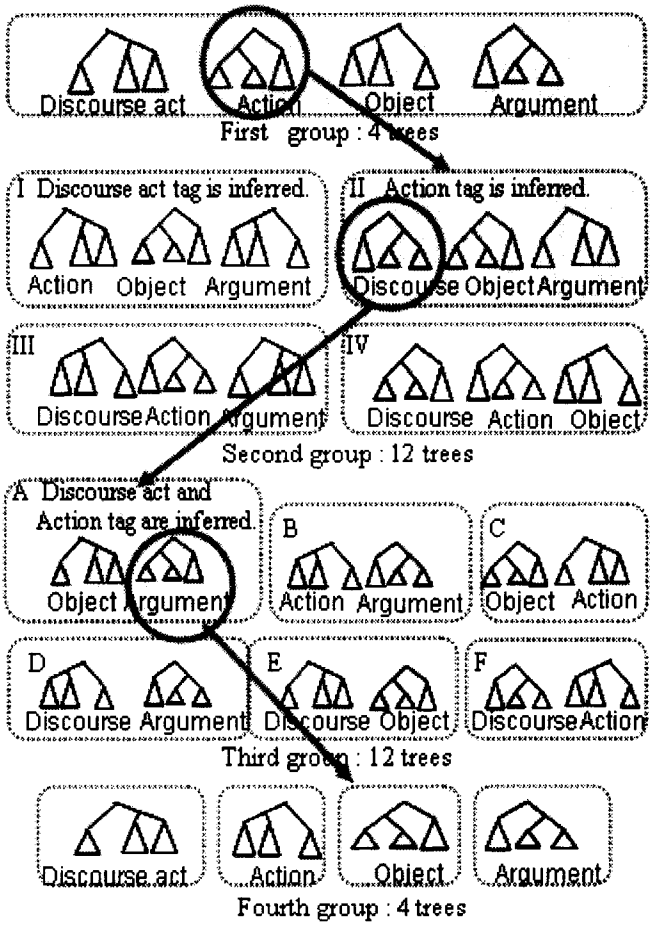


Figure 18-7. 32 decision trees.

4.2 Acquisition of Speaker's Knowledge from Dialogue Data

Various types of restaurant information might be included in restaurant guide dialogues. We employed a dependency structure annotation tool to extract restaurant information from the dialogues and produce a dependency structure for each utterance in them. We then unified all relevant information. Figure 18-8 shows the extraction from utterance No.0032 in Figure 18-4, and the unification of No.0030 and No.0032.

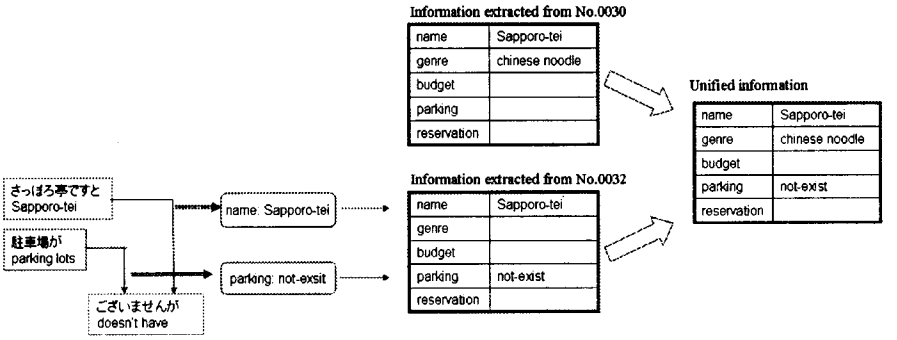


Figure 18-8. Information extraction and unification.

To evaluate the method's validity we conducted an evaluation experiment using 100 restaurant guide dialogues (937 utterances), which contain information on 702 restaurants. The results were a precision of 83.3% and a recall of 64.1%.

5. DEVELOPMENT OF SPOKEN DIALOGUE SYSTEMS

5.1 Corpus-Based Spoken Dialogue System

We constructed a prototype spoken dialogue system as a workbench for evaluating suitable technologies for different parts of the spoken dialogue system[10]. Figure 18-9 the system configuration and the information flow between components. Most of the components in the system are implemented by using the statistical information obtained from the corpus. The speech understanding module calculates the similarity between the input sentence and the example whose previous intention tag is the same as the tag of the preceding system response[11]. To decide an intention of the system's response, we use a trigram of the intention tags created from the corpus with LITs.

We have conducted experiments to evaluate the performance with six subjects using small example data and large example data. From the results of these experiments, the rates of task completion have improved 20% by adding the example data.

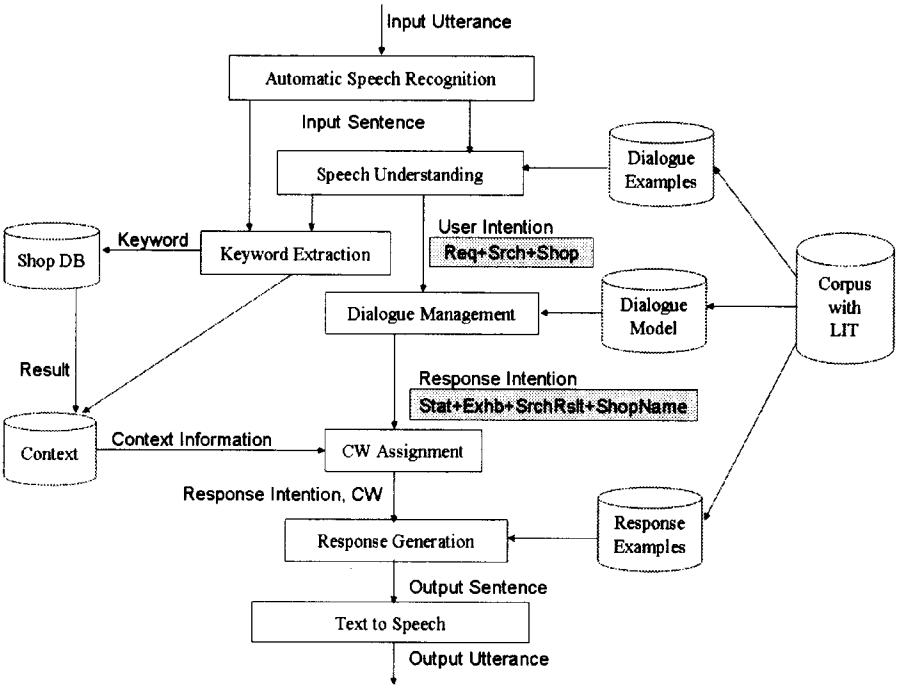


Figure 18-9. Configuration of a prototype spoken dialogue system.

As a future work, to the spoken dialogue system we will consider applying speech intention understanding using decision trees and speech management using dialogue-structural trees.

5.2 Example-Based Spoken Dialogue System

We have proposed a technique that controls spoken dialogue using dialogue examples, and a technique to collect dialogue example data from dialogues performed between a human subject and a pseudo-spoken-dialogue system based on the WOZ scheme[12]. Furthermore, we extended the technique to be able to handle context-dependent utterances[13].

This architecture is named "GROW." As Figure 18-10 shows, a network connects the dialogue system and the WOZ system. Replies created by the dialogue system are transferred to the WOZ system and the Wizard corrects it only if necessary. The data corrected by the Wizard are then sent to the dialogue system, presented to the user and preserved in the dialogue example database. When correction is unnecessary, the reply generated by the dialogue system is presented to the user as-is. This architecture enables the system to automatically add the correct dialogue data as a new example.

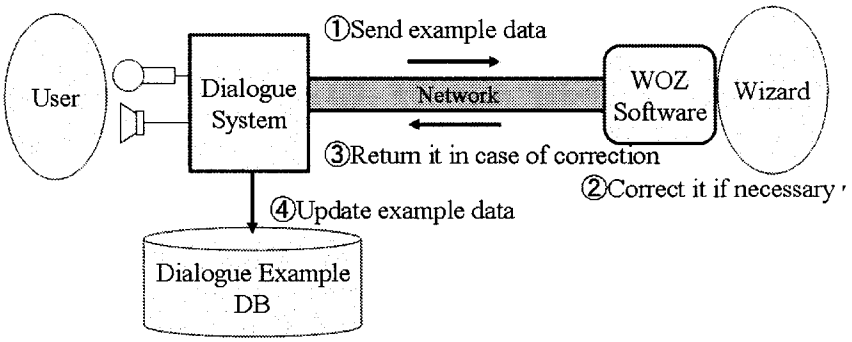


Figure 18-10. Configuration of the "GROW" architecture.

From the results of an evaluation experiment performed with human subjects, we found the success rate of reply generation to improve as the number of the examples increased.

6. CONCLUSION

We have introduced a group of research activities and presented the various findings from the CIAIR in-car speech corpus. All these studies were accomplished primarily by utilizing a large-scale corpus. Generally, since conversation in a car is task-oriented and simpler than general conversation, the in-car spoken dialogue corpus is highly suitable for research on spoken dialogue systems.

In future, we will continue our research using the corpus, the tool, and the prototype system introduced in this paper.

ACKNOWLEDGMENT

This work is partially supported by a Grant-in-Aid for COE Research (No. 11CE2005) and for Scientific Research (No. 15300045) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

REFERENCES

[1] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, Spontaneous speech corpus of Japanese, Proceedings of 2nd International Conference on Language Resources and Evaluation, 947–952 (2000).

[2] N. Kawaguchi, S. Matsubara, K. Takeda, and F. Itakura, Multidimensional data acquisition for integrated acoustic information research, Proceedings of 3rd International Conference on Language Resources and Evaluation, 2043–2046 (2002).

[3] N. Kawaguchi, S. Matsubara, I. Kishida, Y. Irie, Y. Yamaguchi, K. Takeda and F. Itakura, "Construction and Analysis of the Multi-layered In-car Spoken Dialogue Corpus," Chapter 1 in *DSP in Vehicular and Mobile Systems,* H. Abut, J. H.L. Hansen, and K. Takeda (Editors), Springer, New York, NY, 2005.

[5] N. Kawaguchi, S. Matsubara, K. Takeda, and F. Itakura, CIAIR in-car speech corpus - influence of driving states-, IEICE Transactions on Information and Systems, vol. E88-D, no.3, 578–582 (2005).

[6] Y. Irie, S. Matsubara, N. Kawaguchi, Y. Yamaguchi, and Y. Inagaki, Design and evaluation of layered intention tag for in-car speech corpus, Proceedings of International Symposium on Speech Technology and Processing Systems, 82–86 (2004).

[7] T. Ohno, S. Matsubara, N. Kawaguchi, and Y. Inagaki, Robust dependency parsing of spontaneous Japanese spoken language, IEICE Transactions on Information and Systems, vol. E88-D, no. 3, 545–552 (2005).

[8] S. Kato, S. Matsubara, Y. Yamaguchi, and N. Kawaguchi, Construction of structurally annotated spoken dialogue corpus, Proceedings of 5th Workshop on Asian Language Resources, 40-47 (2005).

[9] Y. Irie, S. Matsubara, N. Kawaguchi, Y. Yamaguchi, and Y. Inagaki, Speech intention understanding based on decision tree learning, Proceedings of 8th International Conference on Spoken Language Processing (2004).

[10] K. Hayashi, Y. Irie, Y. Yamaguchi, S. Matsubara, and N. Kawaguchi, Speech understanding, dialogue management and response generation in corpus-based spoken dialogue system, Proceedings of 8th International Conference on Spoken Language Processing (2004).

[11] S. Matsubara, S. Kimura, N. Kawaguchi, Y. Yamaguchi, and Y. Inagaki, Example-based speech intention understanding and its application to in-car spoken dialogue system, Proceedings on 19th International Conference on Computational Linguistics, 633–639 (2002).

[12] H. Murao, N. Kawaguchi, S. Matsubara, Y. Yamaguchi, and Y. Inagaki, Example-based spoken dialogue system using WOZ system log, Proceedings of SIGdial Workshop on Discourse and Dialogue, 140–148 (2003).

[13] H. Murao, N. Kawaguchi, S. Matsubara, Y. Yamaguchi, K. Takeda, and Y. Inagaki, Example-based spoken dialogue system with online example augmentation, Proceedings of 8th International Conference on Spoken Language Processing (2004).

Chapter 19

EXPLOITATION OF CONTEXT INFORMATION FOR NATURAL SPEECH DIALOGUE MANAGEMENT IN CAR ENVIRONMENTS

Markus Ablaßmeier and Gerhard Rigoll
Institute for Human-Machine Communication, Munich University of Technology, Arcisstr. 16, 80333 Munich, Germany, {ablassmeier|rigoll}@tum.de

Abstract: This chapter focuses on the exploitation of context information for a situation- and user-aware dialogue management implemented in a framework for the automotive environment. A generic dialogue manager for driver information systems (like infotainment and communication systems) and driver assistance systems has been developed and tested. One main focus in the development was the ability to make context-dependent decisions. The dialogue manager provides flexible and user-centered speech dialogues and supports multimodal interfaces, like buttons or turning knobs combined with speech. A frame-based approach is used for the dialogue control. The XML description allows an easy specification and overview over the dialogue structure. Visual outputs are monitored on several displays in the car. The evaluation shows an improvement of effectiveness and a higher joy of use through the possibility of submitting several pieces of information in only one dialogue step with natural speech comparing to a menu-based spoken dialogue. The context-dependent information agents reached a high acceptance by the users. The test persons rated the context-based way of frame-based interaction as comfortable and important.

Key words: Context, multimodal, dialogue management, automotive, agents

1. BACKGROUND KNOWLEDGE

This chapter gives an introduction to the research field in the car domain as well as multimodal interaction, and the relevance of context influences are described.

1.1 Driving Aspects

During the few last years, a growing number of functions concerning the information, communication, entertainment and comfort of the driver and passengers have been introduced in the automotive environment. Navigation systems and office applications, such as calendars and phone-books, are just a few of the accessories currently available not only in upper class limousines. These new technological advancements offer the 21st-century driver with many potential benefits. Yet, as a direct consequence of the functional complexity, the interaction with such interfaces is getting more and more difficult for the user which often leads to different kinds of distraction and operation errors. Even the once simple radio system, consisting of a few buttons for the volume and the different radio stations is getting more complex as the driver can now choose a specific song from his/her entire MP3-Collection. As a consequence, standard user input, such as selections from a list or browsing through a matrix menu has to be enhanced. The realization of these human-machine interfaces requires careful development and planning, since social, ergonomic, technical and context aspects have to be considered. Every kind of interaction creates an additional workload to the driver which can – in the worst case – affect the driving performance, the primary task. Because of this, easy-to-use in- and output methods, which do not require high motor activity or a very accurate vision are a must for in-car use. Furthermore, the automotive systems demands non-distracting and fast interaction methods in order to reduce the time of eyes off the road.

Primary Tasks are segmented into navigation, steering, and stabilization. Secondary tasks are operations, like reactions to and dependent on driving demands, but they are not essential to keep the vehicle on track, e.g. honking, gear shifting and fuel check. Tasks not concerning the actual driving itself are categorized as tertiary tasks. If the driver interacts, i.e. with a communication and infotainment system in such a stress phase (tertiary task), inattention, distraction, and irritation occur as a consequence of the high workload resulting from a superposition of the tasks mentioned above, which will become manifest in an increased error potential and in erroneous operations of these tertiary systems.[1] Rassmusen[2] and Donges[3] introduce an in-depth classification of driving tasks.

1.2 Multimodal Input and Output

Multimodal interfaces combine natural input modes – such as speech, pen, touch, manual gestures, gaze, gait, and head and body movements – in a

coordinated manner. Multimodal interfaces are largely inspired by the goal of supporting more transparent, flexible, effective, efficient and robust interaction. The flexible use of input modes is an important design issue. This includes the choice of the appropriate modality for different types of information, the use of combined input modes, or the alternate use between modes. Input modalities can be selected according to context and task by the user or system. Especially for complex tasks and environments, multimodal systems permit the user to interact more effectively. Because there are large individual differences in abilities and preferences, it is essential to support selection and control for diverse user groups. For this reason, multimodal interfaces are expected to be easier to learn and use. The continuously changing demands of mobile applications enable the user to shift these modalities, e.g. in-vehicle applications.[4]

1.3 Context Influences and Knowledge

While steering and stabilizing a car, a lot of influencing factors affect the interaction and, as a consequence, on the dialogue between driver and the tertiary systems to be operated. These factors are summarized as context parameters and can be classified into three main subgroups: environmental, user, and system context parameters.

Traffic volume, road, weather conditions or even the amount of fuel and other car sensors are continuously subject to change. These factors may strongly influence the driver's performance and attention, and are referred to as environmental context.

Also the user her- or himself has a strong impact on the dialogue. An expert who is familiar with the system may prefer other display contents and shortened dialogue structures compared to a novice user. Being on a private journey, the driver will have other needs and preferences compared to a daily routine ride (e.g. the trip to get you to your office). Moreover, emotional expressions of the user play an important role for an effective context-adaptation.

The system context can also influence certain dialogue steps. Some systems state conflict with one another. For example, an intelligent navigation system should disable the menu item "start navigation" unless the driver does provide a destination. Also different knowledge sources can deliver important information for the dialogue (see Section 2).

In dialogue systems, contextual parameters are applied in three different ways: The form of the dialogue can be changed by adapting the verbosity in dependence of the user state. On the other hand, context can be exploited for making information available to the system. Thus, information does not need to be gathered explicitly from the user. For instance, let the car be in Munich, and the driver just feeds in "Arcis Street," then the city can, by default, be concluded from the context for reasons of plausibility. At last, contextual factors can trigger the system to initialize a dedicated dialogue. An example is an active prompt of the system to evade a traffic jam.

2. SPOKEN DIALOGUE SYSTEMS AND KNOWLEDGE SOURCES

A spoken dialogue system is defined as a computer system which uses spoken language to interact with a user. This interaction aims to solve a certain task.[5] Dialogue systems include speech recognition, speech synthesis, language understanding, and dialogue management. The dialogue can have different initiation strategies: In user-driven initiation the user keeps the initiative, however in a system-driven initiation the system keeps the initiative throughout the whole dialogue. In mixed initiative systems the initiative changes throughout the dialogue. According to McTear[5], spoken dialogue systems can be classified into three main types, depending on the methods used to control the dialogue with the user. These are finite state-based, frame-based and agent-based systems.

In the finite state-based approach, the dialogue consists of a predefined sequence of states and conditioned transitions between them. In each state, the system prompts for a user input that generally is expected to be a single word. Depending on this input, the evolving dialogue runs on alternative ways through the dialogue graph. In general, this kind of system is only suitable for clearly structured dialogues with limited quantity and complexity of user input. On the other hand, technical complexity is rather low.

In frame-based systems, the user is asked questions that enable to fill slots in a template in order to perform a task. The dialogue flow is not predetermined, but depends on the user's input and the information the system has to elicit. However, if the user provides more than the re-quested information, the system can accept this information, and check if any additional item of information is required. The dialogue definition consists of system prompts, together with a condition which has to be true for the prompt to be relevant. Referring to the knowledge sources, frame-based systems require an explicitly defined task model because this is used to

determine which question still has to be asked. Frame-based systems provide the possibility to freely decide which and how much pieces of information to enter. This leads to a more natural kind of communication and is essential in case the user doesn't know at the beginning which information is actually needed to succeed.

Agent-based dialogue systems allow complex communication among the system, the user, and the underlying application in order to solve a problem or task. These systems tend to be mixed initiative, which means that the user can take control of the dialogue, introduce new topics, or make contributions that are not constrained by previous system prompts. Concerning these systems, McTear[5] regards communication as an interaction between several agents, each of them capable of reasoning about their own goals and actions. Progressive dialogue context is considered in the dialogue model. In general, there is no given dialogue definition, but the system poses the questions which are required to accomplish the task. To handle this complexity, a huge technical effort is necessary, concerning the dialogue manager as well as pre-processing modules.

The dialogue management has to verify the recognition engine's hypotheses about the user's utterances. The most primitive way is to explicitly ask for an acknowledgment after each input. A much more natural dialogue flow can be achieved by one single acknowledgment of all data in the end of the dialogue. Of course then, the user has to specify in an intermediate step which piece of information has been recognized wrong. The more natural input the dialogue system allows, the more flexible the verification strategy can be handled.

To enable a successful and natural dialogue, the dialogue manager requires knowledge sources. Referring to McTear[5], these sources are called dialogue model. The model might consist of different types of knowledge sources: a dialogue history, a task record, a world knowledge model, a domain model, a generic model of conversational competence and a user model.

3. SYSTEM DESIGN AND FRAMEWORK INTEGRATION

The design of the developed dialogue manager aims to extend an existing multimodal framework (see Section 3.3) by providing natural spoken dialogues and a new approach to driver information, the so-called information agents. This complex system necessitates the management of dialogues and to consider the context aspect.

3.1 Intelligent Agents

The focus of this research field is the situation- and user-aware presentation of information to the driver, the multimodal in- and output, as well as the context-adaptive cross-linking of functions. One approach to these issues is the idea of so-called information agents. They are introduced to offer an optimal situation-dependant support to the driver. She or he should efficiently be led through dialogues matching her or his current according to the context.

Situation-awareness refers to the consideration of external influences. The dialogue between user and system has to be context-dependent (see Section 1.2). The agents can be initiated by the user in case he has any needs, as well as by the system, for example, if the car is running out of gas.

Other examples for this idea of agents are a restaurant guide automatically considering the drivers preferences and an end-of-journey-agent which looks for a parking lot and transmits a map of the surrounding area to the PDA when the driver arrives at the desired destination.

3.2 Input and Output

The dialogue manager is able to process input in terms of intentions. This means, the output string of the recognizer is preprocessed in terms of semantic interpretation. Accordingly, tactile input as well has to be preprocessed. This generic intention based input processing enables to plug arbitrary input modalities and their preprocessing module onto the framework. Multimodal interaction is assured. While running a dialogue, the input modality can freely be changed. A well-designed spoken dialogue per se provides assistance to the driver because there is no need to turn his visual focus from the street to a display.

The frame-based approach for dialogue management is realized and implemented. It provides adequate flexibility while keeping technical requirements manageable. In this case, flexibility means user's flexibility to freely decide how much information to submit in one step. By this, an expert user can reduce the dialogue run, whereas a novice user still can be led through the dialogue step by step. This flexibility as well reduces handling errors because there is no pre-assigned sequence in which the information slots have to be filled.

Natural language in this context is a spoken input where words without semantic relevance may appear. The system is able to process inputs like for example "Well... Please take me to Arcis Street 16 in Munich and take the fastest way." The semantic interpretation is done by a module called EASY discussed in Section 3.3.

As already mentioned, the dialogue should not be controlled only by spoken input, but by other modalities as well. To allow for tactile interaction, a module had to be developed to determine the user's intention by relating tactile input with the current state of the displays.

The dialogue manager's output is freely configurable with respect to the message pattern used in our framework (see Section 3.3). There are three points in the dialogue flow where an output can be defined. First, when successfully finishing a dialogue, messages including user's input information can be sent, for example, to a service, like a navigation system.

Furthermore, it is possible to put out messages after having processed a user input. This can for example be applied to give feedback about the dialogue progress. The third point is the system requesting an input if necessary. Spoken feedback is synthetically generated in our setup. The text to be spoken is sent to a TTS-server via a HTTP-request (See Section 3.3).

One basic principle of usability design is that in any situation, the user has to know the state the system is in. To ensure this, a color-coded speechy-symbol is shown in the head-up display. When finished, the recognized intentions are displayed for three seconds. In case of an error, the user can understand that the malfunction of the system is due to a recognition error and can correct this. In case the confidences of the speech recognizer stay below a customizable threshold, a red speechy-symbol is shown to signalize an error. A prompt will then be repeated more verbosely.

3.3 Framework

The framework is organized in form of a client-server structure, consisting of an input, a fusion, and an output layer. The input layer consists of all kinds of input devices (like a speech recognizer or a button array). In the output layer, there are application modules, like a navigation system or an MP3-player. The core unit is the multimodal integrator. All modules of the input and the output layer must register as clients at the database of the integrator. The communication between input, fusion, and output layer is realized via a bidirectional exchange of string messages streamed over TCP/IP-connections (socket backports).

Using a look-up table, a meta-device (the so-called command mapper) converts all messages of the single recognizers. The mapping process is based on the formalism of a context-free grammar. Thus, the proprietary message output strings of the individual recognizers are formatted into a standardized device-independent structure. The strong modularization allows for a fast and straightforward replacement and an integration of additional modules. Consisting of several networked components, the integrator

interprets the multimodal message stream that is continuously arriving from the individual recognizers.

The input of the recognizers is combined via Late Semantic Fusion (LSF). A string parser checks the messages for syntactical correctness and for integrity. A finite state machine provides and manages the database for the multimodal integration process. In this process, secondary knowledge is included from the application module and the integration status. The intention decoder forms the central component within the multimodal integrator. Considering additional context information (see Section 1.3) the messages are evaluated via a semantic unification process. The result is checked by a set of additional components (e.g. an error manager). Finally, the integration unit generates a device-independent command which is, analog the lines of above, transformed in a proprietary format of the application modules. If, for any reason, the resulting command can not be applied in the current system context, or is incorrect, the dialogue manager generates a dedicated error dialogue.

The following modules were used to integrate the dialogue manager into the existing framework.

Our experimental setup uses an ASR called ODINS developed by our institute[86]. This speaker-independent recognizer is based on intra-word triphone HMMs.

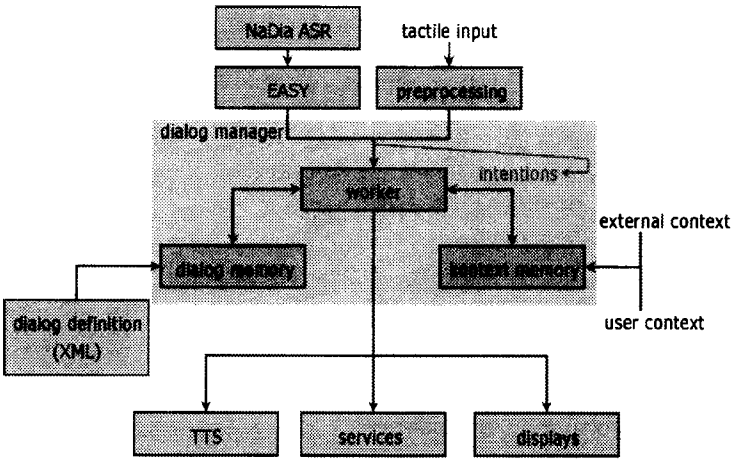


Figure 19-1. Dialogue Manager's Architectural Overview.

[8] Institute for Human-Machine Communication", Technische Universität München

The phoneme models were trained with a corpus containing spontaneous speech utterances from the Verbmobil-project[7].

As input, one has to provide a speech model, a grammar which contains all possible permutations of words as a lattice-network, ideally with probabilities for each word transition. Second, it requires a thesaurus, a collection of all possible words, and one or more phoneme representations of it. ODINS uses the Sampa phonem list.

The recognizer has to be activated manually, for example, by pressing a push-to-talk-button. The recognizer output is an n-best-list containing five sentences with the highest sentence confidence, together with single word confidences, which are the logarithmized probabilities for each word.

In order to be used by the dialogue manager, the results of the ASR have to be interpreted semantically. This is done by another institute's development called EASY[8].

It basically tries to determine user's intentions out of the utterance detected by the recognizer. For example, a spoken input like "I'd like to drive to Munich, to Arcis street, please." would lead to two intentions "destination_city: munich" and "destination_street: arcis street".

EASY uses Bayesian Belief Networks to match the recognized words to one or more predefined user-intentions. To perform this, it requires of course a source of knowledge which defines the composition of the Bayesian Network.

Its outputs, the intentions, are rated by confidences. These are used by the dialogue manager to weigh the input intentions. The speech output in our setup is synthesized by a server running AT&T Natural Voices. The output-string to be synthesized is sent via a HTTP-request to the server which returns a WAV-file.

To be able to use tactile input devices, an additional module have to be designed which preprocesses the input commands. Furthermore, this module is responsible for controlling the displays. The dialogue manager and the dialogue definition should be kept free of display-specific commands. The desired command flow is a meta-command like "show restaurant search" by the dialogue manager which is received and interpreted by the display control. It accordingly sends out the display specific commands. This allows different behaviors on different displays. A central information display realized as touch screen, for example, could show a map with restaurants nearby while in the HUD, these restaurants can be shown as a vertical list.

XML is used to specify the dialogue. Each dialog has an arbitrary number of subdialogues; each of them has a context condition assigned. The subdialogue-node has two kinds of children: A send-element in which socket messages can be defined and the current information frames.

They are associated with intentions coming from the pre-processing input modules. This association is one basic part of the dialogue definition. By this, the appropriate dialogue is figured out and the user's information is processed. A "required"-attribute differs mandatory and supplementary information frames. Only mandatory frames are enquired, optional ones have to have a default value assigned.

Each frame may have different "inquiry"-children which contain information about how to prompt for information to fill this frame. Every inquiry has a "verbose"-value assigned to. Thus, differently detailed prompts can be defined and used by the dialogue manager to enable context adaptation and error management.

3.4 Context Processing

One main focus in the development of the dialog manager was the ability to make context-dependent decisions. As stated in paragraph 1.3, different context parameters have an impact on the driver.

This dialogue manager provides the following possibilities to affect a dialogue:

- It is capable of varying the dialogue itself, as well as initiating a dialogue as a response to a certain situation. Varying the dialogue means, for example, changing the verbosity of the dialogue output with respect to the user's knowledge about the system.
- The initialization of a dialogue would be reasonable if, for example, the car is running out of gas.
- The third way of using context information is to retrieve dialogue input out of it. An example would be the actual city in which the car is placed. This could be used as information for a navigation dialogue which the user doesn't need to provide.
- The appropriate in and output modality according to the context can be used.
- An intelligent error management can avoid or solve errors by the context.

The current version of the dialogue manager uses a rule-based approach for context processing. Context variation is done by defining an arbitrary number of subdialogues where each subdialogue is valid for a dedicated context condition.

Dialogue initialization works quite similar. For each dialogue an initialization condition can be set. If no other dialogue is currently active and one of these conditions becomes true, its specific dialogue will be started.

Dialogue information retrieval is possible because each incoming context value is stored. Out of this memory, desired values can be fetched.

4. EVALUATION

The designed and implemented dialogue manager has been evaluated in a usability experiment in our institute's driving simulator. The goal was to analyze the usability of the system for first-contact users as well as the choice of input modality. Test subjects were asked to handle two of the agents mentioned in Section 3.1, but to focus mainly on their driving performance. The agents could be utilized by natural speech and via a controller.

The evaluation was divided into two parts. At the beginning, the task was to enter a navigation address via speech. In the first step, we let the user decide which input strategy to embark to find out the user's intuitive way to deal with this speech interface. The second experimental part introduced the agents. While driving, the fuel agent was started and informed the driver about the empty tank. The user was expected to successfully run through the agent and to choose one of the suggested gas stations.

Afterwards, the subject should notify the system about being hungry using his own words. This fuzzy input started another agent, a restaurant search. The subject could handle these two agents by speech as well as by a keypad and a touch screen. After these tasks were accomplished, the user's subjective opinion has been acquired by a questionnaire.

5. RESULTS

We acquired 22 subjects, 15 male and 17 female. The mean age was 37.7 years. Half of them stated they have already had experience with speech recognition systems, mainly in the telephony context.

The first task of our experiment indicates a tendency towards a complete input of the navigation destination. 14 of the 22 subjects intuitively used this strategy, eight chose the iterative menu-based way.

After having experienced the two input possibilities, 15 persons now have used the one-step strategy. At a closer look, it emerged that out of the seven people at first using the iterative speech menu none had retained this strategy after getting to know the possibility of a complete input in one step.

In the interview afterwards, the participants rated the possibility of entering more than one piece of information at once as very important and comfortable. The analysis of the time and steps the subjects needed to accomplish the tasks revealed an evident result. The possibility to enter all needed pieces of information at once required only 55% of the time using a speech menu for the same task, and thus, it was much more efficient.

The utterances while using the agents in the second part of the evaluation have been from 100% (yes-or-no question) to 50% (free expression of hunger) command-based.

While entering a navigation destination in the iterative way, 91% of the spoken input was command-based. The complete input at once had a ratio of 77%. The possibility to use natural language has not been used as much as expected regarding to the online survey results.

While using the agents, spoken and tactile input were equally used. Interestingly, no significant distinction between different age groups could be determined.

The agents were widely accepted. On a scale from 1 (which was the best) and 6 (as the worst grade), the fuel agent was rated 1.62 and the restaurant search 1.95. A system initiation was only desired to evade distress, for example, only at very little gas left.

The largest problem for the test subjects was the intuitive operation of the PTT-key. The system expected the user to press the button once and shortly before every utterance. Only three of the 22 subjects did this correctly. A much larger part, six of them, kept the button pressed during the whole input like using a walkie-talkie. The most frequent way to use the button was to press it once to start a dialogue but not to press it if the system poses a question.

6. CONCLUSION

In this chapter, an introduction to the automotive domain in terms of usability and presented a possible classification for spoken dialogue systems is given.

We discussed the design of our context-aware dialogue manager and its integration into the existing framework. A usability evaluation revealed an increase in efficiency and joy of use, enabled by the frame-based approach.

The context-adaptive information agents reached a very high acceptance by the user.

In ongoing and future work, the integration of several multimodal combinations of input and output devices with the dialogue manager is going to be implemented and tested.

As well, one might be to evaluate a statistical approach to adapt dialogue parameters in reaction to context influences. Future research might as well deal with an agent-based dialogue management strategy.

REFERENCES

[1] McGlaun, G. et al., "Kontextsensitives Fehlermanagement bei multimodaler Interaktion mit Infotainment- und Kommunikationseinrichtungen im Fahrzeug." Tagungsband VDI-Fachtagung USEWARE 2004, 22.-23.06.2004, Darmstadt. VDI-Bericht 1837 "Nutzergerechte Gestaltung technischer Systeme"

[2] Rasmussen, Skills, Rules and Knowledge. In: IEEE Transactions, SMC-13 (1983), S. 257–266

[3] Donges, E., "Das Prinzip Vorhersehbarkeit als Auslegungskonzept für Maßnahmen zur Aktiven Sicherheit Maßnahmen zur Aktiven Sicherheit." In: Das Mensch-Maschine System im Verkehr, VDI-Berichte (1992), Nr. 948

[4] Oviatt, S. et al., "Error Resolution during Multimodal Human-Computer Interaction", ICSLP 1996, Philadelphia, in: Proc. Vol.I

[5] McTear, M., "Spoken Dialog Technology: Enabling the Conversational User Interface". In: ACM Computing Surveys 34 (2002), Nr. 1, S. 1–80

[6] Thomae, M. et al., "A One-Stage Decoder for Interpretation of Natural Speech. Institute for Human-Machine Communication", Technische Universität München. 2003

[7] Wahlster, W. Verbmobil: *Foundations of Speech-to-Speech Translation*, Springer, Berlin, 2000

[8] Schuller, B. et al., "Multimodal Music Retrieval for Large Databases", ICME 2004, IEEE Int. Conference on Multi-media and Expo, Taipei, Taiwan

Chapter 20

CROSS PLATFORM SOLUTION OF COMMUNICATION AND VOICE / GRAPHICAL USER INTERFACE FOR MOBILE DEVICES IN VEHICLES

Géza Németh, Géza Kiss, Bálint Tóth
Department of Telecommunications and Media Informatics (TMIT),
Budapest University of Technology and Economics (BME), Budapest, Hungary

Abstract: Two long-term goals of our study has been to develop a standardized communication interface between the mobile device and other onboard systems and to create a parametrical, scaleable user interface, both with voice and graphical user input/output. This chapter describes the main requirements, principles, and aspects of a voice/graphical user interface and of a Bluetooth based communication interface. Requirements and limitations for the implementation of speech synthesis on mobile devices will also be introduced. An SMS-reader application will be presented as a sample application of a mobile device on a vehicle.

Key words: Speech synthesis; mobile devices; scaleable interface; SMS reading

1. INTRODUCTION

Abilities of smart phones and other mobile devices have been improved significantly in the past few years. Mobile phones and PDAs became communication centers with the latest wireless technologies (Infrared, Bluetooth, WiFi, GPRS/EDGE/UMTS, etc.). In addition, they possess numerous favorable features, like enlarging display panel, which allows the development of informative and intuitive user interfaces. Their improved performance enables the implementation of complex application, like speech synthesis or recognition, and other advanced tasks. Mobile devices in the

automotive domain still have less importance, although their advantages could also be used both in intra-car and inter-car communications. In both cases an intuitive user interface is necessary for Human-Computer Interaction (HCI). In this context, it is beneficial and helpful to use speech output/input for basic tasks (Speech User Interface, SUI) complemented with a graphical user interface (GUI) for more complex functions.

A vehicle may be the context or active partaker of several kinds of communication: facilitator of personal conversations, entertainment, it can receive safety/emergency messages from another car or authorities, give navigation information, diagnostics messages, or parts of a multimedia car manual, and many other and not yet foreseen services.

Choosing standard mobile devices as HCI interfaces seems to be the right choice for several reasons. The phone is already a very personal device and a unique identification and billing tool, with a lot of already stored personal information (e.g. names, contacts, calendars). It is owned and carried about by the majority of people, which they also plug into the car's audio system upon entering. Therefore it is well suited to store data for the car user interface also, which is useful because users normally do not like to spend time changing car settings and more than one person may use the car regularly. In addition, a vehicle may be sold to another owner several times during its lifetime. Items of the user interface are language/country dependent, which setting is likely to be the same as that set on the mobile.

The hardware limitations of mobile phones are quickly being lifted, as phones with faster CPUs, increased memory size and more advanced displays/speakers come out. These devices are also more reliable than the average consumer product.

It is also easier to integrate other channels of information using mobile phones, e.g. the notifications of a home alarm system. The board computer may also give information to the phone which can help to reduce cognitive load on the driver, so that an incoming call or an SMS signal will not distract the person during emergency breaking or a quick steering-movement. Such cases are of increasing importance as a consequence of more and more information systems packed in cars.

An argument against use of mobile phones in this way could be that car manufacturers can no longer keep the interface in their own hands, so they sell fewer extras with the cars. To circumvent that they could award 'Certified for (brand)' labels to phones that match their strict criteria (or different types of certificates for different levels of criteria) which are pre-proved to operate well within their automobiles, thus giving them a way of controlling the development of such interfaces, but leaving it to the customer to buy a device with his/her preferred price/performance ratio and to change it as s/he sees fit. Security issues may also arise, but these would need to be

taken care of by car manufacturers anyway. It also means that some engineering effort can be spared on behalf of the car manufacturer, although some already implemented hardware components would be abandoned (e.g. car phone design).

Obviously, it is not possible that all new cars come out with such phone-interface integration possibility at once, as we can see on the example of navigation systems, which have existed for over 15 years but are only becoming wide-spread nowadays. Yet the direction of progress to follow is still to be decided. Several projects work on establishing specifications for telematics, i.e. (in the newer sense of the word) automation in automobiles using software or hardware components. There are several in-car software platforms, including the QNX Neutrino real time operating system [1], Microsoft's Windows CE based mobile platform called "Windows Mobile for Automotive" [2], different real-time Linux variants [3], etc. AMI-C specifications [4] define a uniform set of application programming interfaces (APIs) that enable software developers to write applications that can operate in vehicles. The hardware at present is usually some embedded system, integrating operating systems with special hardware such as the Xilinx PLDs (Programmable Logic Devices) [5].

In Section 2 we discuss the possibilities of creating a general interface between the vehicle's board computer and the mobile phone. In Section 3, we look at issues concerning the text-to-speech part of the SUI. In Section 4, we demonstrate the described concept by depicting an existing mobile phone application developed mainly for use in cars.

2. GENERAL INTERFACE FOR MOBILE DEVICES IN VEHICLES

To realize a personalized standardized communication interface in the automotive domain, speech and graphical user interfaces are required. From the automotive point of view, there must be a standardized communication interface with different service classes and security policies. From the mobile device point of view, the same communication interface should be integrated, and standardized speech and graphical user interfaces are required. Speech input and output are basic aspects of human-machine interaction in cars, as it is dangerous and forbidden in many countries to control by hands and supervise by eyes personal communication devices while driving.

The aim of the present study is to investigate both sides in order to define what is required to develop and implement such a system.

2.1 Automotive Domain

In a vehicular environment the following four main points should be considered:

2.1.1 Sensors and Actuators

Sensors measure the actual value of a parameter. The parameter can be binary (e.g. Boolean: back seat is leaned) or numeric (e.g. volume, temperature, maximum speed) type. Actuators react to human interactions, for example set the preferred position for the driver's seat. There can be two ways for the (preferably wireless) input and output communication of sensors and actuators with the user interface:

1. All the sensors and actuators are wireless (e.g. Bluetooth capable) and they communicate directly with the user interface system. In the case of this solution less wire is needed, although at least one wire for power supply is still required. Furthermore, some systems (e.g. Bluetooth 1.1) employ serial communication, consequently the personalized communication device should connect sequentially to all the sensors and actuators, and in addition this solution is expensive.

2. A much better solution is when all or most sensors and actuators are connected via wires to a control center, which in turn communicates wirelessly with the personalized user interface device. The centralized control of sensors and actuators makes the system extendable, as it will be described in 2.1.4. The major disadvantage is that more wires and a control center are required.

2.1.2 Service Classes

In order to make the system scaleable and usable on different types of cars with different features, the concept of service classes should be introduced. Service classes are divided horizontally into two categories: input and output services. This separation is required, as input and output can be realized with different methods (e.g. input with the buttons on the steering wheel and speech output). Vertically there are different types of service classes, like temperature, seats, Hi-Fi, speed, etc. The latter also have sub-classes (e.g. seats service class has three subclasses: front right, front left, back seat). It is defined individually for every car-type which service classes are supported in keeping with the functions you can control using the personal communication interface.

2.1.3 Security Policy

Different security categories must be defined to prevent the users from reaching service classes in different situations. Basically we have to distinguish three situations: car is not in use, startup, and driving. At least one security category must be defined for all the service classes either to allow or to ban the user from reaching it. For example the driver's seat position can be changed while the car is not in use or during startup, but should not be changed while the user is driving.

Service classes and the rules of security policies may be included in an XML description file. This way new service classes and different security policies can be defined during software update.

2.1.4 Standardized Communication

There are doubts about using wireless (e.g. Bluetooth) in cars, as the lifetime of a car series is about 15-20 years, but we cannot suppose that any given technology will exist in mobile devices so long. It is possible that in 5 years a new technology will replace for example Bluetooth, just as Bluetooth has substituted wired and infrared communication in several situations.

To solve the problem, the control center must have a standardized communication interface. The interface should be independent from the physical medium (e.g. Bluetooth). This way if a new physical level communication standard emerges, only the physical medium connected to the communication interface should be replaced.

Sun's JINI environment [6] realizes a dynamically distributed system that makes the handling of sensors and actuators safe and simple; Service Classes and a Security Policy can be included, and the communication interface layer can also be realized with JINI. JINI is used in the previously mentioned AMI-C environment, which can be a solution for standardized in-car control centers in the future.

2.2 Mobile Device Domain

Mobile devices should use the same communication interface as the control center. To make the personalized communication interface widely applicable, it must be supported by many devices.

Apart from machine-to-machine communication, a standardized human-to-machine interface is also required. Unfortunately a lot of mobile devices do not have public Software Development Kits (SDKs), but for example most Symbian OS [7] and Microsoft Windows Mobile based devices do. These smartphones and PDAs are widely available and are rather cheap. In

addition, the development for these devices is similar to the development for desktop computers. These smart devices have enough computing power to calculate complex algorithms, like those for speech synthesis and even limited vocabulary speech recognition, and the devices have rather large, color display panel, which enables the development of intuitive speech and graphical user interfaces.

Unfortunately, even the two aforementioned mobile operating systems are not compatible with each other, and yet new other systems may also be released anytime. Consequently, a standardized graphical and speech user interface is required.

2.2.1 Standardized Graphical User Interface

The user interface should be rendered in runtime according to a description file that can be common for different mobile platforms (e.g. WinCE and Symbian). The description file includes the user controls (combobox, checkbox, buttons, pictures, etc.), their positions and the action that is performed when the user activates one of them. The description file is realized in XML (eXtended Markup Language) structure. There are four main arguments for using XML:

1. The Speech User Interface can also be included in the description file (see Section 2.2.2.).
2. With the extension of the XML analyzer module functions may be called if the user activates a control.
3. The user interface XML definition file can easily describe the interface components by being subdivided into service class, security policy, etc. sections.
4. The XML realizes a standardized, easy-to-use data environment.

There is an existing solution for the first aspect (i.e. VoiceXML [8]), and there are also tools that realize the second feature, but these technologies are not supported by mobile devices. There are some features that are supported by Windows Mobile (e.g. ASP.NET for Mobiles), but these are not supported by Symbian OS based phones and vice versa. Furthermore the third aspect given above is a very important part of automotive-mobile control and supervision. It should be also implemented in the user interface definition file.

Let us give a simple example for the XML description of a user control in Figure 20-1:

```
<UserControl Name="myTextBox" Type="textbox"
ServiceClass="Temperature" Size="120px" Posx="10" Posy="5"
Input="keys" Input="voice" Output="GUI" Output="SUI"
Action="setTemperature" Security="All">Please define the in-car
temperature</UserControl>
```

Figure 20-1. XML Definition file example.

In this case a 120 pixel wide textbox is rendered at (10, 5), it belongs to the Temperature Service Class, the value can be set with the keys of the mobile or with voice, and Security Policy allows users to set the value anytime. The actual value is presented both vocally and graphically. If the value changes, the "setTemperature" function is called, and the initial content of the textbox and also the text of the vocal prompt is "Please define the in-car temperature".

2.2.2 Standardized Speech User Interface

It is dangerous and in most countries it is also forbidden to control and supervise the mobile device when the vehicle is moving. To make the usage of the mobile device safe, a speech user interface is a viable partaker. The input is realized by speech recognition, and the output is produced by speech synthesis.

It is very critical that a user-independent speech recognition technique should be used, as the training process of user dependent recognizers frustrates drivers and they can easily loose their motivation for using the speech input. The performance of the latest mobile devices is far from being satisfactory for continuous speech recognition, and in case of large fixed vocabularies the calculation time also dramatically increases. If the speech recognizer vocabulary at any time is not more than 150-200 words or phrases, the speed and accuracy of recognition may be acceptable. In the automotive domain this amount may be enough, as in a well designed, intuitive dialog system the number of elements in the vocabulary can be quite few.

Speech generation provides the vocal output of the system. It should inform users about the actual values measured by the sensors, about the possible words and phrases that can be recognized, it should read information messages, etc. Besides unlimited vocabulary Text-To-Speech (TTS) which is of limited quality, specialized very high quality subsystems for well defined topics (e.g. numeric values, dates, etc. [9]) should also be implemented. Users are used to getting very high quality audio from the car speakers and they are not interested in the technical difficulties of speech generation.

Unfortunately, Microsoft's SAPI (Speech Application Programming Interface) and VoiceXML are currently not supported by mobile devices, consequently a standardized speech I/O system should be defined that runs on all major mobile device platforms. Speech generation/synthesis engines should be recompiled for different processors (ARM, RISC, MIPS, etc.) but the speech user interface is to be realized according to the definition file that was shortly introduced in Section 2.2.1. Also a subset of VoiceXML may be implemented and used in the definition file as long as VoiceXML is not supported in mobile devices.

The graphical and the speech user interfaces have to handle service classes and security policies as well. For example users are able to roll the windows, but are not allowed to control breaks with the user interface. This restriction is required because for example the noise of the environment may influence speech recognizer accuracy and it would be dangerous if a word or phrase could be misrecognized for such a command. The same "error" could occur if the user could push a button for breaking.

3. SPEECH SYNTHESIS IN MOBILE DEVICES

Designing user interfaces for mobile devices is quite a challenging task. The size of the input interface (e.g. buttons) is rather small, as well as the ordinary output interface, the screen. In many cases speech output and input can be a smart solution for creating intuitive and easy-to-use user interface for mobile devices.

Speech generation, as one of the main output interfaces, facilitates the use of many mobile applications, although we must face the limitations of mobile devices. Today's mobile devices have limited storage size and processing power from the speech synthesis point of view. The latest research showed that good speech quality and the above-mentioned parameters are in contradiction with each other. Consequently, a compromise must be struck and the optimal ratio between quality, storage size and processing power must be found.

If the chosen mobile device possesses large storage size and fast processor speed, good speech quality can be achieved; namely more technologies can be involved in the module based TTS system (e.g. name and address reader [10], number reader [9], diacritic reconstruction [11], large exception vocabulary, etc.), otherwise the basic speech synthesis technology should be used.

3.1 Porting Speech Synthesis to a Phone Platform

Unfortunately, mobile devices do not have a common programming interface for speech output and input like SAPI (Speech Application Programming Interface), which is available on Microsoft Windows platforms. There is no operating system level interface between the TTS engine and the application. Consequently, each application must refer to a TTS engine directly, e.g. as a dynamically (DLL) or a statically linked library (LIB). The lack of a common speech programming interface prevents speech-enabled applications from being released independent of speech engines as the operating system does not keep a record of text-to-speech engines. Therefore, the license of a TTS engine for mobile devices must always be bought resulting in an increment of the development and the final product costs. In addition TTS vendors realize the programming interface with different functions and different parameters – there is no common interface present for mobile devices. For that reason multilingual support cannot be realized well because if the TTS engine is changed, the name and the parameters of the TTS functions must also be changed in the source code, and in case of different languages the TTS engine often comes from different vendors.

Another problem originates from the different architectures of mobile devices. The audio interface and the main instruction set (e.g. ARMI, MIPS processors, etc) might be different even in case of devices from the same manufacturer. Accordingly, the TTS engine must be ported to all different types of devices on which we would like to run our application.

Realizing the text-to-speech conversion in client-server architecture solves the problems mentioned above, and enables new features as well. For example, client side TTS engines cannot be charged according to their usage, but server side ones can be.

As Figure 20-2 shows, the client sends the text to be read with additional flags (language, quality, character of voice, speed of reading, user's rights, etc.) to the server via a data communication channel. In case of mobile devices it can basically be GPRS, EDGE, UMTS, but in new devices it might even be e.g. WiFi. The server selects the appropriate TTS engine according to the flags and if the user has adequate rights, the text-to-speech conversion is done, and the result is sent back to the client on the same communication channel as an audio stream or as an audio file. Then the client buffers the incoming stream according to the speed of the connection, and plays it. In order to minimize the data flow, it is worth using an audio compression method, like ITU-T G.729, which provides toll-quality speech with only moderate processing requirements and delay times.

The signaling data are sent and received over TCP/IP on port 80 (HTTP, Hyper Text Transfer Protocol) and the audio stream is sent over a predefined port. If the routers do not allow traffic on the second port, the server sends the stream over port 80 as well. The client and the server application can be realized in different programming languages, but they must use the same communication protocol.

JAVA should be a good solution to make the client software platform independent, although even the latest JAVA edition (MIDP 2.0, JSR 118), which is supported by the newest mobile devices, does not have classes for the low level functions. If the application does not require enhanced, low level features, JAVA is a good choice. Otherwise the application must be implemented in the device specific language. In case of Symbian OS based devices it is the Symbian specific C++, and in case of Windows Mobiles it is C# or VB.NET on .NET Compact Framework. At the time of writing, these two types are the most widely spread "smart" mobile devices on the market. With the client-server solution not only the problems of the TTS conversion may be solved on mobile devices, but more features can be added to the main application, like automatic software update.

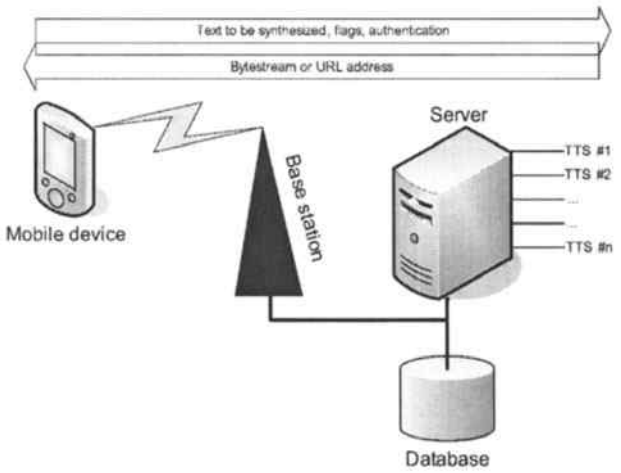


Figure 20-2. Client-server TTS implementation.

3.2 Implications of the Specialties of Mobile Device Hardware: Limitations and Potentials

Times have passed when we had to be ontented with our mobile phone having a small monochrome display, a few built-in utilities, hard memory

limits allowing for a few SMS messages and beeping ring-tones. The latest intelligent mobiles have high resolution color displays, over 30 MBytes of internal memory and possibly more than 1.0 GByte replaceable external memory, the option of downloading and installing various applications, high-speed RISC processors and advanced sound playback functionalities. We do not yet see where this process of development is converging, but the current state of hardware has already opened up the way for a myriad of new possibilities.

The phone is a special device in the sense that it was primarily created to be a voice interface between people, which is reflected even in today's mobiles by the relatively small display and keyboard (motivated by the requirement to retain portability as well), therefore voice should remain the natural interface for human-machine interaction also. Fortunately the aforementioned development of hardware now permits the use of speech recognition and speech synthesis, which together can realize a SUI. Speech recognition can be used for commands and dictating (e.g. SMS and E-Mail messages), Text-to-Speech synthesis can be used for reading out messages, announcing events (e.g. incoming calls) or even voicing the whole menu system.

Nonetheless, the hardware limitations still restrict the range of TTS technologies that can be used: even though the memory available for storing voice data has largely increased, users generally do not want to spend a considerable percentage of it for the voice interface (which they may rightly consider a self-evident part of the phone interface), so corpus-based synthesis [e.g. 12] is not realistic as yet, unless it can be put onto ROM in the factory. Formant synthesis needs very little amount of memory for the voices; this is also favorable if the use of several different voices is desirable for the application. But we also have to consider that its quality is quite limited, and it is computationally quite expensive, so for the time being it may give longer response times and shorter stand-by time, as extensive use of the processor quickly drains the battery. Diphone or triphone synthesis with a limited set of triphones [e.g. 13] yields smaller database sizes and affordable computation time, but the number of voices must be limited. Although newer phones can play sounds at 22kHz or even higher sampling rates, the sampling rate of the recorded speech comprising the database should be kept low (e.g. 8 kHz telephone quality, which people find quite acceptable according to our findings), since a higher sampling rate means almost proportionally higher computation time because of the signal processing required to create the right intonation takes most of the time. Please refer to [14] for more details about optimizing the TTS for the phone.

The GPRS technology has brought the internet and server-based voice-solutions to the phone platform. The 3G technology has made video-calling

possible, which could make another server based application possible that may be especially useful for the deaf and hard-of-hearing and increase the intelligibility and the visual experience for everyone. Namely, supporting the voice of a service by displaying a talking head or a hand-signaling figure calculated on the server-side.

4. IN-CAR APPLICATION: THE SMSRAPPER

There are ways to initiate or receive a phone call without pressing a button or looking at a screen, namely one-touch dialing, voice-dialing, automatic call reception. This can be used effectively in the car so that one does not have to turn his/her attention from driving and lose seconds that could prove to be valuable in an emergency situation. But until recently there was no such solution for receiving SMS messages: one had to stop driving before reading through the messages. The SUI of the phone brings a solution in this aspect also.

The SMSRapper® application, developed jointly by BME TMIT and M.I.T. Systems Ltd. in Hungary is, to our knowledge, the world's first application product that runs on Symbian phones and reads the incoming messages aloud according to the user's preferences. At the time of writing the technology is available for Hungarian, German, Polish, and Spanish. It is already being used by the subscribers of T-Mobile Hungary.

SMSRapper can be used directly as a simple phone application, or the phone could be connected to the car audio system though a Bluetooth connection. This way the car could be adapted to a new user just by connecting his/her phone to the system.

Figure 20-3. In-car application of an SMS-reader.

The application has several settings associated with the phone's profiles (general, in the street, negotiations, silent, etc.) each of which has a default value appropriate for most users, with different behavior in each profile. For example, one can set how fast, how loud and how many times (s)he wants the program to read the message and which properties (s)he wants to hear (e.g. sender, date, time), separately for every mode. Voice with the associated language can be chosen for announcing messages. Language identification from the SMS text can be turned on or off. It can be specified if SMSrapper is to be brought into the foreground when a message arrives.

When an SMS arrives, the program identifies the sender of the message based on the phone-number if it is in the phone's address book and reads his/her name; if not, it reads the number. The language of the message is made known to the user before it is read and this language is chosen for reading the text if it is available; otherwise the default language will be used. For a more detailed description of the application see [14].

Feedback from users of SMSrapper showed that other features are also needed for a more convenient use. When you drive your car, you may not want the device to automatically read out certain texts, e.g. highly confidential business information. Such messages usually come from certain acquaintances of yours, so you may want to put them on a "black-list" which contains the address-book entries whose messages are never to be read. At the same time, you may want that messages from other persons (maybe family members) be read out even if the phone is in silent mode, and put them on the "white-list". A further option is a "grey-list" if you want to be notified about someone's SMS but do not want it to be read aloud. Another extra feature that fits into the application's scope is reading the name of the caller before ringing starts. This way setting (and remembering) personal ringtones, or recording the person's name as a ringtone, can be spared.

5. CONCLUSION

Within the scope of this chapter only a basic sketch of the possible advantages of a standardized integration of mobile devices into vehicles could be given. The authors are open to future co-operation with interested partners in order to more deeply explore this domain.

REFERENCES

[1] S.Ethier and R. Martin, "Instant-on Technology for In-Car Telematics and Infortainment Systems", http://www.qnx.com/download/download/10386/instant-on_mini-driver_whitepaper.pdf

[2] Edward Lansinger, "Windows Mobile for Automotive: A Platform for Smart Telematics Systems", Convergence 2004, Detroit

[3] http://www.realtimelinuxfoundation.org/

[4] Scott J. McCormick, "AMI-C (Automotive Multimedia Interface Collaboration) Fostering Global Communication", ITU-T Workshop on Standarization in Telecommunication for motor vehicles, ITU Headquarters, 2003.

[5] K.M. Parnell, "Reconfigurable Vehicle", http://www.xilinx.com/products/iq/ReconfigurableVehicle02AE-118.pdf

[6] Jini Network Technology, http://www.sun.com/software/jini

[7] http://www.symbian.com

[8] *Voice Extensible Markup Language, v.2.1*, http://www.w3.org/TR/voicexml21/

[9] G. Olaszy, G. Németh, "IVR for Banking and Residential Telephone Subscribers Using Stored Messages Combined with a New Number-to-Speech Synthesis Method", in D. Gardner-Bonneau ed., *Human Factors and Interactive Voice Response Systems*, Kluwer, 1999, pp. 237-255

[10] G. Németh, Cs. Zainkó, G. Kiss, M. Fék, G. Olaszy and G. Gordos, "Language Processing for Name and Address Reading in Hungarian", Proc. of IEEE Natural Language Processing and Knowledge Engineering Workshop, 2003., Beijing, China, pp. 238-243.

[11] G. Németh Cs. Zainkó, G. Olaszy, G. Prószéky, "Problems of Creating a Flexible E-mail Reader for Hungarian", Proc. of Eurospeech'99, Vol. 2, pp. 939-942, Budapest, Hungary, 1999

[12] W.N. Campbell, "CHATR: A High-Definition Speech Re-Sequencing System", Proc. of 3rd ASA/ASJ Joint Meeting, 1996, pp. 1223-1228

[13] G. Olaszy, G., Németh, P. Olaszi, G. Kiss, G. Gordos, "PROFIVOX - A Hungarian Professional TTS System for Telecommunications Applications", International Journal of Speech Technology, Kluwer Acedemic Publishers, Volume 3, Numbers 3/4, December 2000, pp. 201-216

[14] G. Németh, G. Kiss, Cs. Zainkó, G. Olaszy, B. Tóth, "Speech Generation in Mobile Phones", in D. Gardner-Bonneau and H. Blanchard eds., *Human Factors and Interactive Voice Response Systems*, 2nd ed., Springer, forthcoming

Chapter 21

A STUDY OF DIALOGUE MANAGEMENT PRINCIPLES CORRESPONDING TO THE DRIVER'S WORKLOAD

Makoto Shioya[1], Takuya Nishimoto[2], Juhei Takahashi[3] and Hideharu Daigo[3]
[1]Systems Development Laboratory, Hitachi, Ltd., 1099 Ohzenji, Asao-ku, Kawasaki-shi, Kanagawa 215-0013, Japan; [2]Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan; [3]ITS Center Research & Planning, Japan Automobile Research Institute (JARI) 1-30 Shiba-daimon 1-chome, Minato-ku, Tokyo 105-0012, Japan

Abstract: We have conducted a study during the fiscal years 2000-2002 concerning a network-distributed voice-activated telematics service system and another study in 2003-2004 concerning a voice-activated system and driver distraction. Based on those original studies, this paper presents dialogue management corresponding to the driver's workload and other factors, with the aim of help to develop consensus for voice-activated in-vehicle systems.

Key words: Dialogue management; telematics service; voice-activated system; driver's workload.

1. INTRODUCTION

Drivers are opting to operate their in-vehicle systems ever more frequently these days in order to access car navigation service and other telematics services that are being made possible by the ongoing progress of Intelligent Transport System (ITS) technologies. Accordingly, the operating ease of an in-vehicle system is one of the critical issues involved in efforts to improve driver comfort and convenience while assuring safe vehicle operation.

Voice-activated car navigation systems have been implemented in production vehicles by applying voice recognition and voice synthesis technologies. However, using voice-activated controls in situations involving a heavy driving workload, such as when turning at intersections, passing or merging with traffic, entails the possibility of causing driver's workload. One conceivable way of reducing that possibility would be to suspend voice interaction when there is a heavy driving workload. However, if the criteria for suspending voice interaction differed among the manufacturers of car navigation systems or among the systems themselves, it could affect safety and might detract from the convenience of telematics services.

Against that backdrop, the authors have conducted a study during the fiscal years 2000-2002 concerning a network-distributed voice-activated telematics system and another study in 2003-2004 regarding a voice-activated system and driver distraction. Based on the results of those studies, this chapter describes the aspects that should be shared or develop consensus among car navigation equipment manufacturers and among different systems with respect to dialogue control in voice-activated systems designed to reduce driver's workload. Furthermore, it presents the results of an examination of dialogue management principles and compliance procedures in dialogue control corresponding to the driver's workload, as considered from the standpoint of voice technology researchers.

2. NETWORK-DISTRIBUTED VOICE-ACTIVATED SYSTEM

Our research envisioned a network-distributed voice-activated system[12] capable of providing over a network the various types of information that drivers need, in a timely manner and without interfering with driving operations. Studies have been conducted for the purpose of developing what could be called a "voice-activated driving partner," which supports conversational interaction while discerning the driver's circumstances.

Toward that end, one objective was to optimize the collaboration and functional distribution between an in-vehicle system that uses voice recognition/synthesis technologies and the network. Figure 21-1 shows a conceptual diagram of such a network-distributed voice-activated system. Furthermore, it was confirmed on the basis of experiments that a voice-activated in-vehicle system needs to incorporate dialogue control capability that takes into account driver's workload and information presentation functionality that considers the priority level of information.

The following insights were gained through this research concerning a safe human-machine interface (HMI):

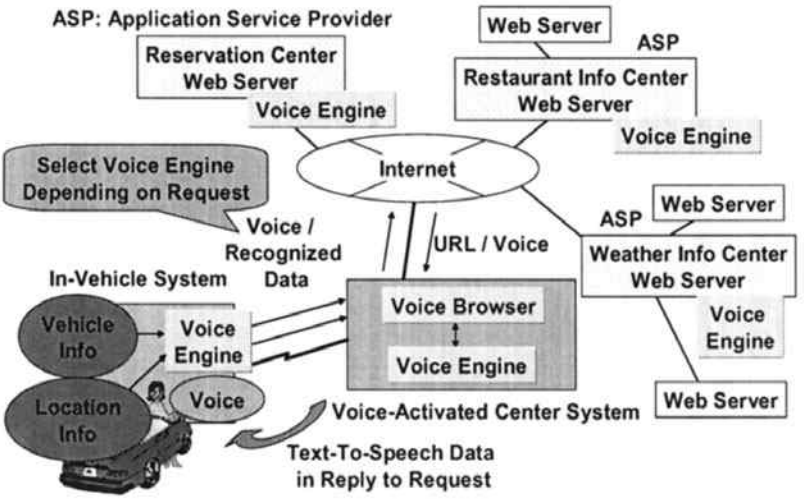


Figure 21-1. Proposed Network-Distributed Voice-Activated Telematics Service System.

1. Commonality of voice commands and sound effects as well as dialogue control corresponding to the driver's workload are effective in reducing impacts during driving.
2. Information must be presented to drivers in a way that considers the priority level, and it is important to strike a balance between the timing for presenting information and its priority.
3. Items 1 and 2 above differ depending on the driver's driving skill and degree of proficiency with a voice-activated system. It is necessary to have a function for adaptively varying the response of a voice-activated system to match such factors.

3. VOICE-ACTIVATED SYSTEM AND DRIVER DISTRACTION

The results of the aforementioned studies confirmed the necessity of having dialogue control that takes into account driver distraction[3-5].

Accordingly, it was decided to continue with research that included the possible consensus of voice-activated technology, and for that purpose, a study was launched concerning voice-activated systems and driver distraction. This work was conducted by the Voice Telematics Working Group, consisting mainly of voice-related researchers who engaged in 3

limited Japanese electronics equipment manufacturers and the research of a network-distributed voice-activated system, which was newly formed under the Mobile Systems Committee that was established to support standardization activities at the Japan Automobile Research Institute (JARI).

The following objectives were determined for this research project, taking into account the possible consensus of voice-interaction control and other measures for reducing impacts during driving.

1. To make clear the items that should be shared by voice-activated in-vehicle systems
2. To prepare a draft proposal of the requirements for voice-activated systems

4. TENTATIVE PROPOSAL FOR DIALOGUE MANAGEMENT

As the first step, desirable dialogue management in the use of voice activation was provisionally examined, based on the insights gained in the research concerning a network-distributed voice-activated system.

4.1 Scope of Dialogue Management Principles and Compliance Procedures

Dialogue management principles and compliance procedures were examined within the following scope.

1. Dialogues between the driver and an in-vehicle telematics system while a vehicle is in motion.
2. In such dialogues, voice is used as the main medium for inputs from the driver to the in-vehicle system. Voice inputs are primarily recognized by using voice recognition technology.
3. A synthesized voice or sounds are used as the principal media for conveying the outputs of the system to the driver in such dialogues.
4. The in-vehicle telematics system not only provides traffic information, points of interest (POI) information, news and weather reports, it also supports various popular Internet services such as e-mail, e-commerce, information searches and entertainment.
5. The aim of the study concerning dialogue management principles and compliance procedures is to show examples of the basic principles, requirements and recommendations with respect to accessing telematics services within the scope of the driver's spare

mental capacity after processing the driving workload (i.e., without the occurrence of any driver's workload).

4.2 Basic Principles and Requirements with Regard to Dialogue Management Proposed in This Study

4.2.1 Proposed Principles

A critical fundamental principle of dialogues for offering telematics services is that first priority is given to the processing of the driving workload as the primary task. This principle for putting priority on the processing of the driving workload can be explained as follows:

- Priority is given to the processing of the driving workload, so long as the driver always has sufficient mental capacity for handling the workload involved.
- The driver should keep his/her mind on driving until the processing of the driving workload is finished. In other words, so long as attention is not directed to things other than driving, priority is given to the processing of the driving workload.

Or the principle can also be expressed as follows:

- If the driver engages in a dialogue for accessing a telematics service (secondary task) within the scope of the person's spare mental capacity after having processed the ever-present driving workload, then priority is being given to the primary task.

Figure 21-2 shows one example of the concept of dialogue control corresponding to the driver's spare capacity.

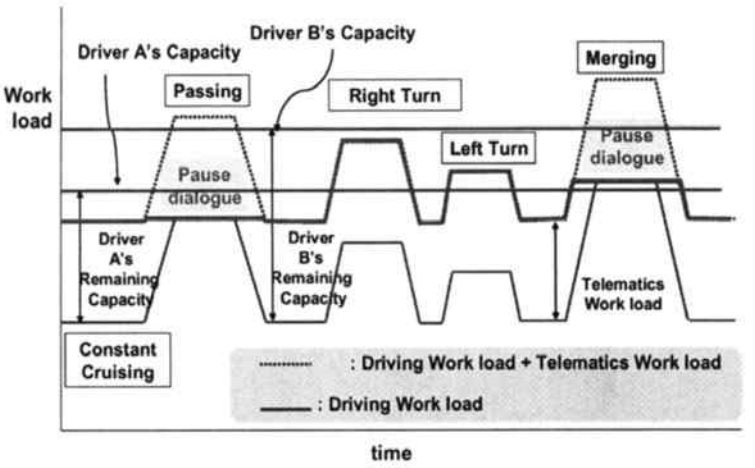


Figure 21-2. An Example of Dialogue Control for Driver B.

4.2.2 Proposed Requirements

The following aspects were considered as fundamental requirements for complying with the basic principles.

- Dialogues for accessing telematics services must be controlled such that the driver's capacity for processing the workload involved in driving is secured first, and the workload for obtaining the telematics service is processed within the scope of the driver's remaining spare capacity.
- The workload involved in driving and the workload for accessing a telematics service should either be estimated in advance or measured in real time.
- The driver's capacity for processing these workloads should either be estimated in advance or measured in real time.
- Dialogue control should be executed in a way that takes into account the driver's state as well as various constantly changing driving conditions such as the road environment, traffic conditions, vehicle status, trip status and so on.
- Dialogue control should be executed in a way that gives consideration to the priority level of various types of services.
- The driver should be informed in a suitable manner of the status of the in-vehicle telematics system.
- The driver should have ultimate control over the dialogues.

4.3 Requirements of In-vehicle Telematics System Proposed in This Study

4.3.1 System Configuration

The basic premise is that the use of telematics services should be controlled in situations involving a heavy driving workload in order to ensure safe vehicle operation. In other words, in situations where the driving workload plus the telematics service workload is equal to or greater than the driver's capacity to handle these workloads, access to telematics services is controlled.

The control procedure takes into account adaptive functionality whereby control is performed in a manner that depends on each driving situation and the capacity of individual drivers.

4.3.2 Proposed System Requirements

The essential elements making up an in-vehicle telematics system are listed below.

1. A means of conveying indexes of the driving workload, telematics service workload and the driver's capacity for handling them to the system.
2. A means of judging situations where the driving workload plus the telematics service workload is equal to or greater than the driver's capacity to handle these workloads.
3. A means of conveying the situation to the driver.
4. A means of controlling the telematics service workload depending on the situation and the driver's capacity.
5. The elements above should allow for personal customizing through selection and adjustment according to the individual driver's driving skill and degree of proficiency with a voice-activated system.

In order to improve the transparency of the system to drivers, it will be important to:

a) make the system easy to understand,
b) make it easy for drivers to become accustomed to the system, and
c) include a tutorial.

4.3.3 Recommended System Functions

The in-vehicle system should incorporate the following functions in order to meet the basic requirements.

1. A means of detecting various driving situations.
2. A means of judging the possibility of driver's workload occurring.
3. A means of conveying the situation to the driver.
4. A means of controlling the telematics service workload in a manner that depends on the situation and the driver's capacity.

Specific examples of these functions and their applications are described below.

1. The following are among the possible means of detecting various driving situations.

   a) Examples of methods of detecting the driving workload index
   - Through the detection of driving operations by using information on the driver's operation of the brake pedal, accelerator pedal, steering wheel, turn signals and other devices.
   - By using information output by in-vehicle equipment such as a speed sensor, accelerometer, angular velocity sensor, radar unit, camera or other devices.
   - By using position information or map information such as information on the road geometry, road grade, road width, traffic restrictions (maximum speed, one-way traffic, no right turn, etc.) or road type (city street, school zone street, expressway, etc.).
   - By using the time of day or weather conditions such as the outputs of a road surface sensor, sunlight sensor, raindrop sensor, windshield wiper sensor, headlight sensor, cross wind sensor or other devices.

   b) Examples of methods of detecting the telematics service workload index
   - By different types of service such as according to the respective index for entry and other operating procedures and the difficulty of the dialogue involved in route guidance, parking searches, restaurant searches, payment, e-mail, online quizzes, karaoke, music title searches and so on.
   - By using the telematics information tags such as the respective index for traffic restrictions, accident/disaster information, congestion information, route guidance and other items.

   c) Examples of methods of detecting the driver's capacity
   - Based on age.

- Based on the number of years since obtaining a driver's license or the number of years of driving experience.
- Based on the number of years of accident-free driving.
- Based on a self-assessment such as a declaration of one's level of driving ability, motor skill and information processing ability and a ranking of HMI ability.
- Based on physiological abilities such as by having various physiological functions measured at the time of driver's license renewal and by using those self-declared values.

d) Examples of ways of obtaining temporary self-assessments from drivers

- By having a driver perform a barge-in task by voice input or touch screen input and using the result to temporarily revise reported data on the person's capacity.

2. Examples of methods of judging the possibility of driver's workload occurring:

a) Based on the results of situation detection.

- Calculation and judgment of whether the driving workload plus the telematics service workload is equal to or exceeds the driver's capacity to handle the workloads

3. Examples of methods of conveying the situation to the driver, assuming that voice and sounds are the main media used:

a) For conveying recommended actions to the driver from the system

- Use of voice announcements according to the situation.
- Use of sound effects or musical effects according to the situation.
- Basic separation into about three levels (utterance accepted, request for utterance, system malfunction) which are advised to the driver.
- Designing of easy-to-understand prompts in terms of the number of words (number of syllables), speed and voice range.
- Designing of easy-to-understand voice announcements in terms of the number of words (number of syllables), speed and voice range, and provision of a backup function for advising the driver of the system status.
- The timing and interval of a recommended action should match each type of situation in order to be effective.

b) It is desirable to convey information to drivers via a multimodal means using sound effects, voice, images (videos, photos, graphics, colors, etc.), text or other media suitable to the situation.

- If the media are also to be used while a vehicle is in motion, each one should be expressed in a way that is easy for drivers to understand. Visual, auditory and tactile means should be used suitably depending on the level of emergency or priority.

c) Means of conveying information to drivers should take into account the ranking of HMI devices.

- Because the ability to handle HMI devices differs among individuals, the devices should be ranked and information conveyed to drivers accordingly. HMI devices should be ranked according to the information processing levels of each user age group, including older drivers, and their proficiency with the processing procedures.

4. Examples of methods of controlling telematics service workloads depending on the situation and the driver's capacity:

a) Provision of functions for controlling operations or dialogues according to the situation

- Controlling and limiting dialogues depending on the level of risk, emergency or priority.

b) Details of dialog control

- System should possess a function for interrupting a dialogue when the driving workload becomes heavy and for resuming the dialogue again when the workload becomes lighter. The system should have functions for controlling operations and dialogues separately according the nature of the utterances.
- Several modes of dialogue interruption should be provided, including instantaneous interruption and interruption at a natural break in an utterance. Dialogue sentences and styles should be convertible to ones that are easily understood by drivers, including the use of key words, main points, summaries, menus, headings, full text, chapters, sections, pages and paragraphs.
- Visual, auditory and tactile means should be used suitably depending on the level of emergency or priority. Presentation timing should take into account ease of understanding by drivers. The level of consistency (inconsistency) with the content of other information should be indicated. The system should display the symbols and icons for indicating congestion, road construction, travel time and information.

c) Dialogue control commands

- Approximately, twelve different types of voice commands, for which there is 99% recognition, should be prepared. For in-vehicle systems in Japan, voice commands should be in both Japanese and English. Voice commands should be provided that enable drivers to control the progress of a dialogue.

d) Dialogue control taking into account the ranking of HMI devices

- Drivers' capacity to handle HMI devices varies from individual to individual and also depending on personal preferences and the situation. Accordingly, it should be possible to control dialogues according to individual capacities to handle HMI devices, based on a ranking of the devices.
- It should be possible to select an HMI device level that matches user profiles, including personal preferences.
- It should be possible to select an HMI device level corresponding to the situation. It should be possible to make adjustments within HMI device levels according to user profiles and the situation.

e) Consideration of task success rates and reduction of user dissatisfaction until task accomplishment in the case of HMI devices based primarily on voice interaction.

- Total task time required to complete a task should be reduced as much as possible, as should the total number of steps required.
- Voice recognition rate should be improved. Synthesized voice quality should be enhanced. Response should be improved.
- Number of operations, number of voice recognition failures and number of recognized candidates should all be reduced as much as possible[3].

5. POSSIBLE INTERNATIONAL STANDARDS RELATING TO THIS STUDY

A survey was made of international standards related to dialogue management with the aim of developing a dialogue control standard for voice-activated systems designed to reduce driver's workload. The results made the following points clear.

1. There are relatively few standards that directly govern voice-based dialogue control, but there are many standards related to visual aspects.
2. The standard that is the closest to dialogue control is ISO 15005[6], and the title and scope of application of this standard seem to be identical to what we are considering in the sense of dialogue management in the telematics field.

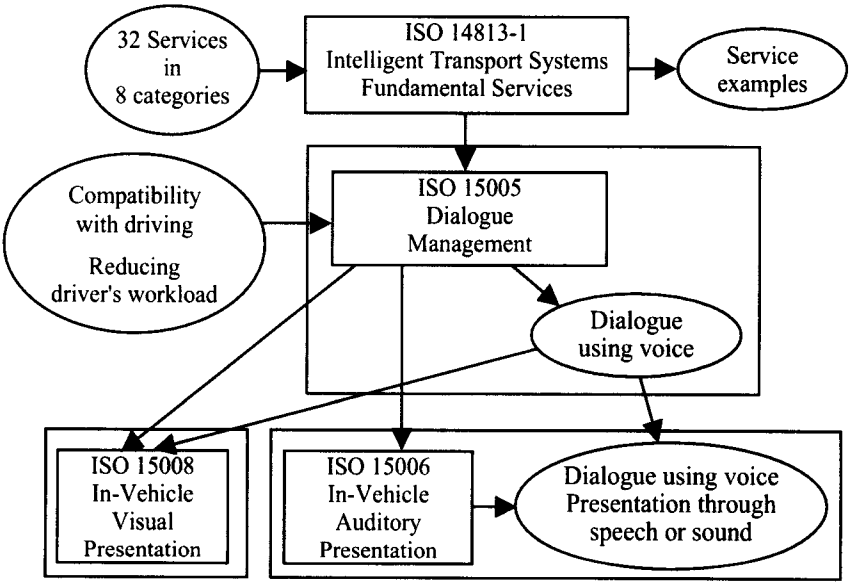


Figure 21-3. Typical HMI-Related ISO Standards.

The correlation among the four international standards that were surveyed is outlined in Figure 21-3. The standards are mentioned in the boxes, and the arrows indicate quotations or directions to take for references. The ISO 15005 standard is organized around the aim of reducing drivers' workloads due to the operation of an in-vehicle system so as to achieve compatibility between telematics service use and driving.

It is thought that a dialogue management standard, including a voice interaction standard as well, could be developed by simply complementing the dialogue principles of ISO 15005 as follows:

citation from ISO 15005:
5.2.4.3.4 Recommendation
TICS should provide timely visual information to the driver.

An example of a supplemental recommendation:
5.2.4.3.4-1 Recommendation

Voice-activated telematics systems will provide the driver with timely audible information.

Example 1: In situations where the driving workload increases such as when turning right or left at an intersection, the driver will be advised sufficiently in advance of the maneuver so that the voice-based telematics service dialogue can be performed safely.

Figure 21-4 shows an example of dialogue control during the execution of a right turn, which is one situation involving a heavy driving workload. The dialogue is interrupted and resumed based on the On/Off status of the turn signal lever, respectively. "Interrupt" and "resume" commands are also provided so as to give precedence to the driver's judgment.

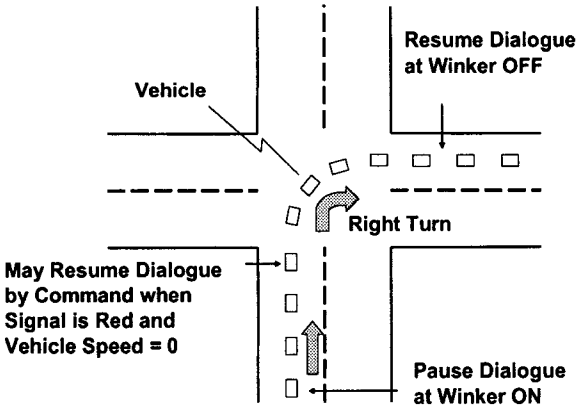


Figure 21-4. Dialogue Control in Case of Right Turn.

6. CONCLUSION

In this study, we have attempted to clarify various aspects that should be shared or consensus needs to be developed among car navigation equipment manufacturers and among different systems with respect to dialogue control in voice-activated systems designed to reduce impacts during driving. Desirable dialogue management corresponding to the driver's workload

were also examined and tentatively proposed from the viewpoint of voice technology researchers. This study has been done as part of a comprehensive study toward the development of consensus for voice-activated telematics systems.

It is also important to choose suitable methods for measurement of workload for voice-user interface systems[7] and to assess the appropriateness of our proposal.

It is hoped that this chapter will serve as a first step toward collaborative activities between parties involved with automotive HMI technologies and those involved with electronics-related voice technologies, with the aim of developing voice activation consensus for original and after-market in-vehicle systems.

ACKNOWLEDGMENT

This paper compiles the results of research conducted by the Voice Telematics Working Group under the Mobile Systems Committee at the Japan Automobile Research Institute. The authors would like to express their deep appreciation to all of the members involved in this project.

REFERENCES

[1] Association of Electronic Technology for Automobile Traffic and Safety, Feasibility Study Concerning the Infrastructure for Network-distributed Voice-activated In-vehicle Telematics Systems, a study commissioned by The Mechanical Social Systems Foundation, March 2003.

[2] Makoto Shioya, et al. Research on Network-Distributed Voice-Activated System Architecture for Telematics Services. 10th ITS World Congress, Madrid, 2003.

[3] Japan Automobile Research Institute, Mobile Systems Committee, Voice Telematics Working Group, A Study of Voice-activated Systems and Driver Distraction, March 2005.

[4] T. Nishimoto, Symposium on Ergonomics and the Usefulness of Mobile Phones and Car Navigation Systems, Japan Ergonomics Society, pp.125-128, Kyoto, March 2004 (in Japanese).

[5] Makoto Shioya, et al. Study on Reference Models for HMI in Voice Telematics to meet Driver's Mind Distraction. 11th ITS World Congress, Nagoya, 2004.

[6] ISO15005 Road vehicles - Ergonomic aspects of transport information and control systems - Dialogue management principles and compliance procedures.

[7] Takuya Nishimoto, Motoki Takayama, Haruaki Sakurai, Masahiro Araki: Measurement of Workload for Voice User Interface Systems, Systems and Computers in Japan, Volume 36, Issue 8, pp.81-89, May 2005.

Chapter 22

ROBUST MULTIMODAL DIALOG MANAGEMENT FOR MOBILE ENVIRONMENTS

Jeonwoo Ko[1], Fumihiko Murase[1], Teruko Mitamura[1], Eric Nyberg[1], Nobuo Hataoka[2], Hirohiko Sagawa[2], Yasunari Obuchi[3], Masahiko Tateishi[4] and Ichiro Akahori[4]

[1] *Language Technologies Institute, Carnegie Mellon University, USA*
[2] *Central Research Laboratory,* [3] *Advanced Research Laboratory, Hitachi Ltd., Japan*
[4] *Research Laboratories, DENSO CORPORATION, Japan*

Abstract: This chapter describes three aspects of mobile dialog management: robustness in the presence of recognition errors; dynamic behavior based on user context (e.g. network connectivity, location); and efficient scenario description for multimodal dialogs. We describe algorithmic techniques for these three aspects of mobile dialog management, and results from empirical user studies are discussed which indicate significant improvement in performance and user satisfaction when these techniques are deployed in a dialog system.

Keywords: Dialog system; mobile environment; dialog management; error handling; context-awareness; multimodal interaction

1. INTRODUCTION

In the past few years, several important studies have appeared on mobile dialog systems in pedestrian and automotive environments (Pellom et al., 2001; Buhler et al., 2002; Minker et al., 2004). Such systems include a navigation interface (to provide route guidance) and remote data access (to provide information on points of interest, tourism sites, weather, restaurant and entertainment facilities, etc.). In mobile environments, dialog systems must provide robust dialog management in order to handle recognition errors and network loss, and to support robust multimodal interactions. Our recent

work has focused on three aspects of robust dialog management in mobile environments.

First, speech recognition errors are more frequent in mobile settings due to noise in the environment, which can make it difficult to support smooth speech dialog communication. To address this problem, we propose a grammar-based error handling approach that dynamically generates correction grammars based on the dialog history, and uses these correction grammars to detect and repair recognition errors. Experimental results show that our error handling technique increased task completion rate while reducing the number of user turns.

Second, network connectivity is not stable in a mobile environment (e.g. the user goes through a tunnel or climbs up a mountain) and the dialog manager should be robust enough to handle changes in network status. When an information request is interrupted by a loss of network connectivity, the user should have to request the same information again when network connectivity is reestablished. The system should maintain a dialog history and automatically restart any pending dialogs as soon as the network becomes available, so that the user does not need to monitor the network status and then ask for information again. Such behavior is an example of a *context-aware* service (Dey and Abowd, 2000), which recognizes contextual events (e.g. a change in network status) and responds by providing relevant information to the user. We designed three types of context for mobile dialogs, and conducted an evaluation of their use in both laboratory and driving environments. Evaluation results show that context-aware dialog management improved user performance and user satisfaction.

Third, mobile dialog systems involve multimodal interactions using both speech and display/touch screen. This implies that dialog scenarios must contain detailed specifications about what to present in each modality (e.g., what to display, what to say, etc.) for each dialog step. In our previous work (Nyberg et al., 2002), we developed ScenarioXML, an extension to VoiceXML which supports flexible dialog switching. We recently extended ScenarioXML to support multimodal interactions, so that dialog developers can easily define a new multimodal dialog scenario for a new domain.

The remainder of this chapter is structured as follows. Section 2 introduces the CAMMIA dialog system, which is framework for our dialog research. Section 3 explains how correction grammars are created based on the dialog history, and how correction grammars detect and repair system errors. In Section 4, we describe context-aware features in mobile dialog systems and discuss the related issue of handling user interruption. Section 5 describes the extended ScenarioXML for multimodal dialog management. Valuable conclusions are drawn in Section 6.

2. CAMMIA

CAMMIA (Conversational Agent for Multilingual Mobile Information Access) is a multilingual spoken dialog system which provides route guidance and information services in English and Japanese. The latest version of CAMMIA was successfully demonstrated at the ASRU 2005 workshop[9].

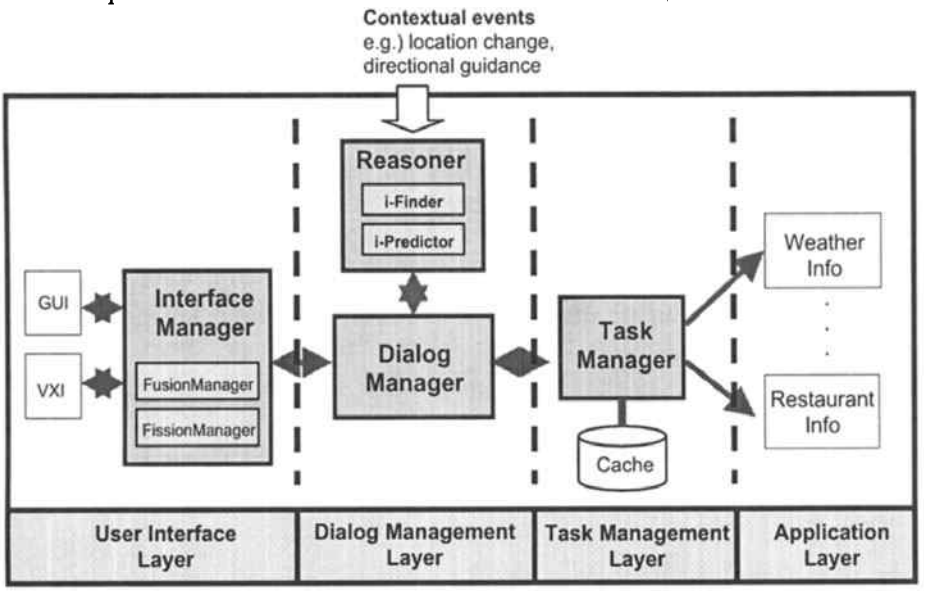


Figure 22-1. CAMMIA multi-layer architecture.

CAMMIA consists of four layers to support multi-modal, mobile information access (Figure 22-1).

- **User interface layer**: This layer is responsible for low-level interaction with the user via a variety of input/output modalities; initially it incorporates a Voice XML interpreter and a tactile interface. The Interface Manager consists of the FusionManager and the FissionManager. The FusionManager converts user inputs from different interfaces into a dialog object that is sent to the Dialog Manager. The current CAMMIA system does not support media fusion, but when incorporating media fusion, the FusionManager will be extended. When

[9] ASRU (Automatic Speech Recognition and Understanding): http://www.asru2005.org

receiving the output from the Dialog Manager, the FissionManager generates an interface-dependent XML for each interface and sends each XML to the proper interface.

- **Dialog management layer**: This layer consists of two subcomponents: the Dialog Manager and the Reasoner. The Dialog Manager represents and manages multiple ongoing topics of conversation. Dialogs are represented as *dialog scenarios*: each scenario consists of a set of states and transitions between states. The i-Finder module in the Reasoner uses contextual events and domain-specific rules to search for information that is appropriate to the user's context (e.g. nearby restaurants, sightseeing attractions and rest areas). The i-Predictor module decides the proper time to provide that information to the user.
- **Task management layer**: This layer explicitly maintains each information download request as a separate task object. When the network is not available, it monitors the network status and downloads the requested information after the network becomes available later. In addition, it supports the use of predictive caching, where the most commonly recurring information requests are autonomously triggered by the system to refresh the local information store. This capability can be used for location-aware suggestions. Predictive caching helps to maximize utility of the system when the network is unavailable, but it does lead to an increase in average bandwidth utilization depending on the size of the cache and the update frequency.
- **Application layer**: This layer includes any additional code and/or data that is required to service information requests from the Task Manager. Typical application layer components include relational database adapters and web service client modules. Such components act as "wrappers" which provide information to the CAMMIA architecture from remote legacy services.

Modular separation of these layers makes it possible to support different types of mobile devices. Small devices like PDAs may have limited system resources and only the user interface layer may reside on the client side. In contrast, mobile devices with more resources (e.g. systems for automotive environments) can be equipped with the dialog management layer (and task management layer), which enables the user to speak to the system in order to manage car devices and controls even when the network is not available.

3. ROBUST DIALOG MANAGEMENT

As speech recognition errors are inevitable in mobile environments, the Dialog Manager should be sufficiently robust to detect and repair errors. CAMMIA addresses this problem with correction grammars (Sagawa et al., 2004a). Dialog Manager maintains a grammar history to store the previously used grammars, and dynamically creates correction grammars using this history and a template (possible patterns for the correction utterances). Three rules are applied to create correction grammars (Table 22-1).

Table 22-1. Rules to create correction grammars.

Rule 1: Copy the grammar rules in the history
Rule 2: Insert the rules in the history into the template
Rule 3: Insert slot-values extracted from the history into the template

When correction grammars are used to recognize a user utterance, the Dialog Manager transits into the "correction of errors" state and repairs errors (task transition errors or slot value errors). For a task transition error, the new task in the correction utterance is initiated. For a slot value error, the old slot value is replaced with the new value in the correction utterance. When the new value is the same as the old one, the second candidate in the n-best list from the recognizer is used, given the assumption that the recognizer failed to recognize the correction utterance again.

Experiments were done to test our error handling approach. A total of 1200 dialog instances were collected from experiments with and without error handling functionality. The analyzed results on those dialog instances show that our error handling increased the take completion rate by 8.1% and reduced the number of user turns by 20.5%.

Another experiment was conducted to compare three types of confirmation styles (explicit, implicit and final confirmations) in terms of usability (Sagawa et al., 2004b). The experimental results with thirteen participants show that in the normal situation, participants preferred final confirmation because it involves fewer user turns. On the other hand, participants preferred explicit confirmation when recognition errors happened, mostly because they want to confirm each slot one by one. These findings indicate that combining final and explicit confirmation provides the best dialog management strategy.

4. CONTEXT-AWARE DIALOG MANAGEMENT

Dialog systems in mobile environments must notice changes in the user's environment to be effective in dynamic situations. For example, the SmartKom Mobile system adapts route guidance based on user preference and location (Malaka et al., 2004). Such systems are described as *context-aware systems* and have been studied in the field of human-computer interaction (Schilit et al., 1994; Dey and Abowd, 2000). We incorporated context-awareness into our system with three context-aware features.

4.1 Context-awareness in mobile dialog systems

- **Network context for robust task management**: The network is not stable in mobile environments. As the CAMMIA system can represent the dialog history and automatically restart the dialog when the network becomes available, the user does not need to monitor the network status and ask for information a second time. At certain intervals, the system can automatically prune obsolete dialogs based on the current user context (e.g. location, elapsed time).
- **Dialog context for flexible dialog switching:** Robust reference resolution is important in flexible dialog switching. As can be seen in Table 22-2, CAMMIA infers search parameters and the estimated time to the destination from the context (S4) when the user changes the dialog topic (U4). In addition, CAMMIA can provide different information to the user for the same question. For example, the utterances U2 and U5 ask for the price. For utterance U2, the system provides price information for both of the places. On the other hand, to answer utterance U5, the system automatically goes back to the sightseeing dialog from the weather dialog because the weather dialog does not support price information; the system then provides price information only for Echo Valley, which is the current dialog context.
- **User context for location-aware suggestions**: When traveling to a location not previously visited, the system adapts to the new environment by suggesting useful information based on the current location and user profile. For example, the system may suggest popular tourist sites near the current location or signal unavailability of the destination based on the estimated arrival time and the location's business hours. The user can modify system behavior by setting user preferences, for example, by adding parking lot availability as a feature that should always be the checked.

4.2 Support for context-awareness

When suggesting new information to the user, the system may interrupt an ongoing user activity or dialog, and this should be carefully designed to minimize user distraction. This raises three interesting design questions: what to say to the user, when to say it, and how to say it (Ko et al., 2005a).

Table 22-2. Example dialog for robust reference resolution. (U: user, S: system)

U1: I am looking for a ski resort in Nagano[10].
S1: There are two famous ski resorts in Nagano. Echo Valley and Tangram Ski Resort.
U2: Can you tell me the price?
S2: Echo Valley is $30 for a one-day pass. Tangram is $27 for a one-day pass.
U3: How far is Echo Valley?
S3: It is 190 kilometers away.
U4: Can you tell me the weather?
S4: **You'll arrive there around 1:30 p.m. The weather forecast for that time is sunny.**
U5: Can you tell me the price again?
S5: Echo Valley is $30 for a one-day pass.

The Reasoner decides what information to suggest due to contextual changes. It has two sub modules: the i-Finder, which searches for relevant information, and the i-Predictor, which estimates interruptibility. When the i-Finder retrieves information relevant to user situation, it stores the information in a pending output queue. When the i-Predictor is invoked, it accesses the pending information queue and determines whether it is a good time to interrupt the user. If so, the information is sent to the Dialog Manager to trigger (or re-start) the appropriate dialog with the user. If not, information is left on the pending information queue. When information in the queue is no longer valid because the context has changed (e.g., the user moved to another location), the i-Finder removes it from the queue.

The Dialog Manager and Interface Manager work together to adapt their behaviors for more efficient interactions and minimum user distraction.

4.3 Evaluation

Preliminary experiments were conducted in a laboratory environment (Hataoka et al., 2005). The ten participants (five males and five females) were asked to use two different systems: a system configured with context-aware functionalities, and a system configured without those capabilities.

[10] Nagano is a prefecture in Japan.

The user tasks were to find a nearby Italian/Japanese restaurant with special constraints such as cuisine and parking lot availability (for location-aware suggestions), find weather information when the network in not stable (for robust task management) and search for sightseeing sites in Shizuoka for picking oranges, apples and strawberries (for flexible dialog switching). As shown in Table 22-3, the tasks required fewer turns when using a context-aware system.

Table 22-3. Average number of user turns.

Task	System	Male	Female
Location-aware suggestion	Context-unaware	30.8	25.6
	Context-aware	26.4	17.2
Flexible dialog switching	Context-unaware	10.0	14.4
	Context-aware	5.6	7.8
Robust task management	Context-unaware	5.0	6.2
	Context-aware	2.0	3.0

Similar experiments were done in a driving environment with sixteen participants (four females and twelve males). The participants were asked to complete their tasks while maintaining a simulated speed of 100 km/h. The goal of the experiments was to determine whether context-awareness is easy to use and helpful for task completion in a driving environment and how much it affects driving behavior (Ko et al., 2005b). The experimental results show that context-awareness significantly improves user performance: the time to complete the tasks decreased on average by 49%, and the number of user turns decreased 68%. After each task, participants completed a user satisfaction questionnaire. The analysis of the questionnaire shows that user satisfaction improved in terms of ease of system use (+52%) and system helpfulness during tasks (+65%).

We also measured the effects of the dialog systems on driving behavior, especially focusing on whether adding context-awareness to the dialog system might reduce the degree of user distraction during driving. For this study, three driving conditions were compared: driving-only baseline, driving while using the context-unaware system, and driving while using the context-aware system. The analysis shows that average car speed was not significantly different, but people tended to change speed more frequently when using the context-unaware system than when using the context-aware system ($p<0.03$). These results indicate that context-awareness could reduce the degree of user distraction during driving.

5. MULTIMODAL DIALOG MANAGEMENT

In our previous work (Nyberg et al., 2002; Obuchi et al., 2004), we developed ScenarioXML to facilitate complex dialog scenario creation. ScenarioXML allows dialog designers to write abstract descriptions for dialog scenarios which are compiled by a ScenarioXML compiler to create VoiceXML which supports dynamic content.

```
<ScenarioXML>
 <grammar>
   <name>weather.gad</name>
 </grammar>
 <state name="ask_weather">
  <slots>
    <field>weather_date</field>
    <field>weather_area</field>
  </slots>
  <prompt>Please tell me the area and date</prompt>
  <screen template="list"/>
 </state>
 <state name="ask_date">
  <slots><field>weather_date</field></slots>
  <prompt>Please tell me the date</prompt>
  <screen template="list">
    <item>today</item>
    <item>tomorrow</item>
    <item>weekend</item>
  </screen>
 </state>
 <state name="ask_area" exp_confirm="yes">
  <slots><field>weather_area</field></slots>
  <prompt>Please tell me the area</prompt>
  <screen template="list">
    <item>Tokyo</item>
    <item>Nagoya</item>
    <item>Karuizawa</item>
  </screen>
 </state>
 <state name="show_result">
  <prompt>It is %result in %weather_area %weather_date</prompt>
  <screen template="info">
    <item>highest</item>
    <item>lowest</item>
    <item>precipitation</item>
    <item>image</item>
  </screen>
 </state>
</ScenarioXML>
```

Figure 22-2. Example of the extended ScenarioXML for weather dialog scenario.

We have now extended ScenarioXML to support a screen interface. Figure 22-2 shows an example of the extended ScenarioXML. It consists of two main tags: <grammar> and <state>. <grammar> defines the grammar name to cover a dialog scenario. <state> is composed of the field name(s), the voice prompt and screen information including a screen template name and items to be displayed on the screen. *%result* in prompt tag refers to dynamic information from the database.

ScenarioXML can easily support explicit, implicit and final confirmation. For explicit confirmation, "*exp_confirm*" attribute can be used. When this attribute is specified "yes", the Dialog Manger will explicitly confirm the slot. Implicit confirmation can be part of the prompt. For example, the prompt can be "For *%weather_area*, what time would you like to know the weather?". The Dialog Manager will replace *%weather_area* with the filled value. The final confirmation is supported when inserting additional state in front of *show_result* state.

```
public class WeatherDialog extends Dialog {
   public WeatherDialog(){
      super("Weather");
      grammarName = "weather.gad";
      needSelection = false;

      Constraint con1 = new Constraint("weather_date");
      con1.setScreenValues("today;tomorrow;weekend");
      con1.setTemplateName("list");
      constraints.put("weather_date", con1);

      Constraint con2 = new Constraint("weather_area");
      con2.setScreenValues("Tokyo;Nagoya;Karuizawa");
      con2.setTemplateName("list");
      con2.setExpConfirmation("yes");
      constraints.put("weather_area", con2);

      Result res = new Result("weather_result");
      res.setScreenItems("highest;lowest;precipitation;image");
      res.setTemplateName("info");
      result.put("weather_result", res);
   }
}
```

Figure 22-3. Example of Java class compiled by the ScenarioXML Compiler.

ScenarioXML is compiled into Java classes for the Dialog Manager and templates for language generation by the ScenarioXML compiler. Figure 22-3 shows an example of the compiled Java class. It inherits from *Dialog* class which defines three main dialog states and dynamic state transitions (Figure 22-4):

- Search state: ask the user to fill the minimum search constraints and display the search results (e.g. There are three Italian restaurants near hear. Luca, Italiano and Bravo)
- Search refinement state: the user can narrow down the searched items with different search options such as price and distance (e.g. which one is closer?)
- Selection state: automatically check context information such as availability (based on the estimated time and business hours) or items that user set in the profile (e.g. credit card acceptance). And then add the selected place to the navigation map.

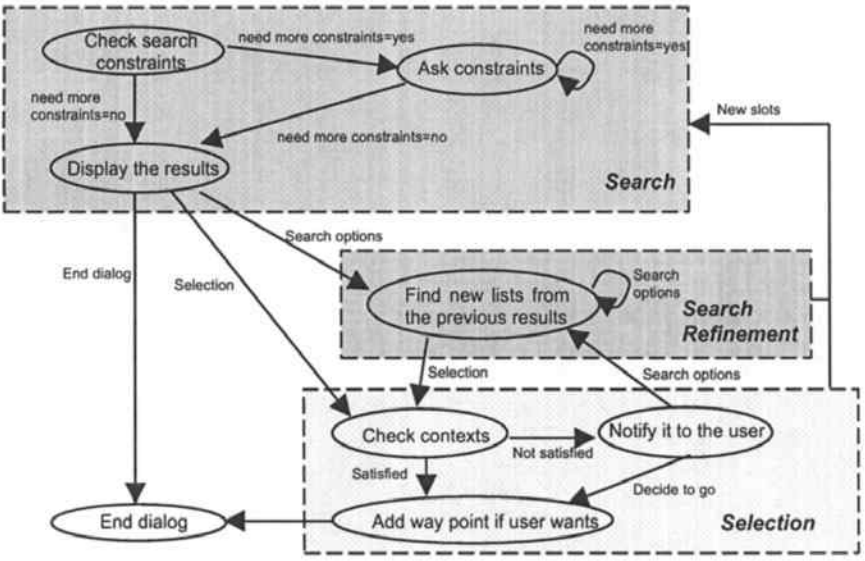


Figure 22-4. Dynamic state transition in "Dialog" class shown in Figure 22-3

As the *Dialog* class supports general transition among the states, the Java classes compiled from ScenarioXML include only upper-level descriptions of constraints and screen objects. The extension of ScenarioXML for multimodal interfaces is still ongoing.

6. CONCLUSION

In this chapter, we have attempted to describe three aspects of mobile dialog management: robust dialog management for error handling, context-aware dialog management in dynamic situations, and multimodal dialog management. For robust dialog in the presence of recognition errors, we have proposed an approach using dynamic correction grammars. For context-aware dialog management, we have described contexts which can be used to adapt the behavior of mobile dialog systems. For efficient multimodal dialog management, we show how we are extending ScenarioXML to support high-level dialog scenario description.

We are currently extending our ScenarioXML compiler to compile the extended ScenarioXML, and investigating how to determine relevant content and system timing for dynamic user environments.

REFERENCES

[1] Buhler, D., Minker, W., Haubler, J., and Kruger, S., 2002, "Flexible Multimodal Human-Machine Interaction in Mobile Environments," In Proceedings of International Conference on Spoken Language Processing.

[2] Dey, A. K. and Abowd, G. D., 2000, "Towards a Better Understanding of Context and Context-Awareness," In Proceedings of Workshop on The What, Who, Where, When, and How of Context-Awareness, at Conference on Human Factors in Computing Systems.

[3] Hataoka, N., Sagawa, H., Obuchi, Y., Tateishi, M., Akahori, I., Ko, J., Murase, F., Mitamura, T., and Nyberg, E., 2005, "Robust Speech Dialog Management System for Mobile and Car Applications," in *DSP in Vehicular and Mobile Systems,* H. Abut, J. H.L. Hansen, and K. Takeda (Editors), Springer, New York, NY, 2005.

[4] Ko, J., Murase, F., Mitamura, T., Nyberg, E., Tateishi, M., and Akahori, I., 2005a, "What, When and How to Communicate: Strategies for Context-Aware Dialogs," Unpublished manuscript.

[5] Ko, J., Murase, F., Mitamura, T., Nyberg, E., Tateishi, M., and Akahori, I., 2005b, "Evaluating a context-aware spoken dialog system in mobile environments," Unpublished manuscript.

[6] Malaka, R., Haeussler, J., and Aras, H., 2004, "SmartKom mobile: intelligent ubiquitous user interaction," In Proceedings of Conference on Intelligent User Interface.

[7] Minker, W., Haiber, U., Heisterkamp, P., and Scheible, S., 2004, "The Seneca Spoken Language Dialogue System," Speech Communication, Vol. 43(1–2).

[8] Nyberg, E., Mitamura, T., Placeway, P., Duggan, M., and Hataoka, N., 2002, "Dynamic Dialog Management with VoiceXML," In Proceedings of the Human Language Technology Conference.

[9] Obuchi, Y., Nyberg, E., Mitamura, T., Duggan, M., Judy, S., and Hataoka, N., 2004, "A Robust Dialog Management Architecture Using VoiceXML for Car Telematics Systems," in *DSP in Vehicular and Mobile Systems,* H. Abut, J. H.L. Hansen, and K. Takeda (Editors), Springer, New York, NY, 2005.

[10] Pellom, B., Ward, W., Hansen, J., Hacioglu, K., Zhang, J., Yu, X., and Pradhan, S., 2001, "University of Colorado Dialog Systems for Travel and Navigation," In Proceedings of the Human Language Technology Conference.

[11] Sagawa, H., Mitamura, T., and Nyberg, E., 2004a, "Correction Grammars for Error Handling in a Speech Dialog System," In Proceedings of the Human Language Technology Conference.

[12] Sagawa, H., Mitamura, T., and Nyberg, E., 2004b, "A Comparison of Confirmation Styles for Error Handling in a Speech Dialog System," In Proceedings of International Conference on Spoken Language Processing.

[13] Schilit, B., Adams, N., and Want, R. 1994, "Context-Aware Computing Applications," In Proceedings of Workshop on Mobile Computing Systems and Applications.

Index